BrightStor™ CA-ASM2® Backup and Recovery

RSVP Reference Guide

4.2

Computer Associates™

This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

This documentation may not be copied, transferred, reproduced, disclosed or duplicated, in whole or in part, without the prior written consent of CA. This documentation is proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, licensed users may print a reasonable number of copies of this documentation for their own internal use, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the confidentiality provisions of the license for the software are permitted to have access to such copies.

This right to print copies is limited to the period during which the license for the product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to return to CA the reproduced copies or to certify to CA that same have been destroyed.

To the extent permitted by applicable law, CA provides this documentation "as is" without warranty of any kind, including without limitation, any implied warranties of merchantability, fitness for a particular purpose or noninfringement. In no event will CA be liable to the end user or any third party for any loss or damage, direct or indirect, from the use of this documentation, including without limitation, lost profits, business interruption, goodwill, or lost data, even if CA is expressly advised of such loss or damage.

The use of any product referenced in this documentation and this documentation is governed by the end user's applicable license agreement.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227-7013(c)(1)(ii) or applicable successor provisions.

© 2002 Computer Associates International, Inc. (CA)

All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

Contents

Chapter 1. Introduction 1-1
1.1.1 $RSVP Command 1-2
1.1.1.1 Executing the $RSVP Command in Batch Mode 1-2
1.1.2 RSVP Controls 1-4
1.1.3 Keyword Defaults 1-4
1.1.4 Terminology 1-5
1.1.5 Pattern Masking of Data Set Names 1-6
1.2 Using This Guide 1-9
1.3 Summary of New Features for Version 4.2 1-10
1.3.1 9999 Files Tape Support 1-10
1.3.2 SMS Support Enhancements 1-10
1.3.3 ISPF Interface Enhancements 1-10
1.3.4 IXR Enhancements 1-10
1.3.5 CA-ASM2 Workstation 1-11
1.3.6 $DASDMNT 1-11
1.4 Documentation Changes 1-12
1.4.1 Removed 1-12
1.5 Reading Syntax Diagrams 1-13

Chapter 2. Reporting 2-1
2.1 Input Selection/Filtering Phase 2-2
2.1.1 Source Selection 2-2
2.1.1.1 Command Format for Source Selection 2-3
2.1.1.2 Keyword Descriptions for DASD Volumes 2-4
2.1.1.3 Keyword Descriptions for IPC and System Catalogs 2-5
2.1.1.4 Keyword Descriptions for SMS 2-6
2.1.1.5 Keyword Descriptions for Transaction Data 2-6
2.1.1.6 Keyword Descriptions for CA-1 TMC and CA-Dynam/TLMS VMF 2-7
2.1.1.7 ISPF for Input Source Selection 2-8
2.1.2 Data Set Selection 2-9
2.1.2.1 Data Set Name Selection 2-9
2.1.2.2 Command Format for Data Set Name Selection 2-10
2.1.2.3 ISPF for Data Set Selection 2-11
2.1.2.4 Attribute Selection 2-12
2.1.2.5 Command Format for Attribute Selection 2-14
2.1.2.6 ISPF for Attribute Selection 2-16
2.2 Sort Phase 2-17
2.2.1 Command Format for Sort 2-17
2.2.2 ISPF for Sort 2-19

2.3 Output Format Phase 2-20
2.3.1 Command Format for Output Format 2-21
2.3.1.1 ISPF for Output Format 2-31
2.4 VSAM Reporting 2-32
2.4.1 Command Format for VSAM Elements 2-32
2.4.1.1 ISPF for VSAM Selection 2-34
2.5 User Exit Descriptions 2-35
2.5.1 Command Format for User Exits 2-35
2.5.2 Sample Input File for EXIT1 2-37
2.5.2.1 Control Statements 2-37
2.5.3 ISPF for User Exits 2-38
2.6 Examples 2-39
2.6.1 DASD Volume Selection Examples 2-39
2.6.1.1 Example 1 - Listing a User's Data Sets 2-39
2.6.1.2 Example 2 - Using IF/AND/ANDIF/ORIF Keywords 2-41
2.6.1.3 Example 3 - Using Data Set Name Selection Keywords 2-42
2.6.1.4 Example 4 - Using Data Set Name Simple Pattern Masking . . . 2-44
2.6.1.5 Example 5 - Listing VSAM Activity Statistics 2-45
2.6.2 SMS Selection Examples 2-46
2.6.2.1 Example 1 - Listing All Volumes in SMS Storage Groups 2-46
2.6.2.2 Example 2 - Listing All SMS Data Sets 2-47
2.6.2.3 Example 3 - Listing All Data Classes in the SMS Database . . . 2-48
2.6.2.4 Example 4 - Listing All Management Classes in the SMS Database 2-49
2.6.2.5 Example 5 - Listing All Storage Classes in the SMS Database . 2-50
2.6.2.6 Example 6 - Listing SMS Data Classes and Storage Classes . . . 2-51
2.6.3 OSCAT Selection Examples 2-52
2.6.3.1 Example 1 - Listing Space Information for Cataloged Disk Data Sets 2-52
2.6.3.2 Example 2 - Queueing for Archival Data Sets With Specific High-level Node 2-53
2.6.4 ASM2CAT Selection Examples (CA-ASM2) 2-55
2.6.4.1 Example 1 - Listing All Unloaded Data Sets in the CA-ASM2 Catalog 2-55
2.6.4.2 Example 2 - Listing All Data Sets Unloaded to the Disk Staging Area 2-57
2.6.4.3 Example 3 - Listing All Data Sets Backed Up to the Disk Staging Area 2-59
2.6.4.4 Example 4 - Listing All Data Sets Unloaded to a Specific DSA Volume 2-60
2.6.5 TMC Selection Examples (CA-1) 2-61
2.6.5.1 Example 1 - Searching for Active Data Sets Created on a Specific Drive 2-61
2.6.5.2 Example 2 - Searching the CA1 TMC for Exceptional Conditions 2-62
2.6.5.3 Example 3 - Listing All Tape Volumes in the TMC Eligible for Reuse 2-63
2.6.5.4 Example 4 - Listing All Files on Tapes Whose Volser Starts with TST03 2-64
2.6.5.5 Example 5 - Listing Files by Tape Volser and File Sequence Number 2-65
2.6.6 VMF Selection Examples (CA-Dynam/TLMS) 2-66

2.6.6.1 Example 1 - Listing All Volumes in the VMF with the Scratch Indicator Set . . . 2-66
2.6.6.2 Example 2 - Searching the VMF Using the Selection Keyword CONTAIN . . . 2-68
2.6.6.3 Example 3 - Listing Active Tape Data Sets Created by a Program 2-70
2.6.6.4 Example 4 - Using the IF/ANDIF Keywords to Search the VMF 2-71
2.6.6.5 Example 5 - Listing All Tapes Used for Payroll Showing Their Scratch Status . . . 2-72
2.6.6.6 Example 6 - Listing All Tapes Not in Scratch Status . . . 2-73
2.6.7 Combined Selection Examples . . . 2-74
2.6.7.1 Example 1 - Searching More Than One Primary Input Source . 2-74
2.6.7.2 Example 2 - Searching the VMF and IPC in a Single Pass . . . 2-76
2.6.7.3 Example 3 - Searching DASD Volumes and the IPC in a Single Pass . . . 2-77
2.6.8 Sort Examples . . . 2-78
2.6.8.1 Example 1 - Using EXTSORT . . . 2-78
2.6.8.2 Example 2 - Using Descending Sort . . . 2-79
2.6.9 Output Format Examples . . . 2-80
2.6.9.1 Example 1 - Using the PRINT Keyword to Change Formats . . 2-80
2.6.9.2 Example 2 - Collecting Space Management Information . . . 2-84
2.6.10 VSAM Reporting Examples . . . 2-88
2.6.10.1 Example 1 - Using VSAMCAT to Reference a DDNAME . . . 2-88
2.6.10.2 Example 2 - Using VSAMCAT to Reference a VSAM Catalog 2-89

Chapter 3. Preprocessing . . . 3-1
3.1 Transaction Files . . . 3-3
3.1.1 Command Format for Transaction Files . . . 3-3
3.1.2 Generating Transaction File Commands . . . 3-6
3.2 Archive and Backup Queues . . . 3-7
3.2.1 Command Format for Queues . . . 3-7
3.3 Controlling Unregistered Data Sets . . . 3-9
3.3.1 Command Format for Unregistered Data Sets . . . 3-10
3.4 Examples . . . 3-12
3.4.1 Transaction File Examples . . . 3-12
3.4.1.1 Example 1 - Generating $SM Commands to Release Space . . . 3-12
3.4.1.2 Example 2 - Generating $SM Commands to Release, Compress 3-15
3.4.1.3 Example 3 - Generating TSO DELETE Commands . . . 3-17
3.4.1.4 Example 4 - One-Step CLIST Technique . . . 3-18
3.4.1.5 Example 5 - Reporting SMS Information from the IPC . . . 3-19
3.4.2 Archive and Backup Queues Example . . . 3-21
3.4.3 Unregistered Data Sets Examples . . . 3-22
3.4.3.1 Example 1 - Delete Unregistered Data Sets . . . 3-22
3.4.3.2 Example 2 - Generating IEHPROGM SCRATCH/UNCATLG Statements . . . 3-26
3.4.3.3 Example 3 - Generating IEHPROGM SCRATCH Statements . . 3-28

Chapter 4. DASD Billing . . . 4-1
4.1 Billing . . . 4-2
4.1.1 Command Format for Billing . . . 4-5
4.1.2 Billing DD Statements . . . 4-6

4.2 Two-Step Billing 4-11
4.2.1 Sample JCL for Two-Step Billing 4-11
4.3 One-Step Billing 4-16
4.3.1 Sample JCL for One-Step Billing 4-16

Chapter 5. Tailoring 5-1
5.1 JCL 5-2
5.2 RSVP User Exits 5-4
5.2.1 $RSBILL ($RSVP Command Billing) 5-4
5.2.2 $RSEXT1 5-4
5.2.2.1 Exit Router Facility 5-5
5.2.3 $RSEXT2 5-7
5.2.4 $RSEXT10 (RSVP Sample &XO Control Statement Processor) 5-8
5.2.5 $RSEXT11 (RSVP Empty Data Set Identification Exit) 5-8
5.2.6 $RSEXT12 (RSVP Non-Empty Data Set Exclusion Exit) 5-8
5.3 User CSECT 5-9
5.3.1 Command Format for User CSECT 5-9
5.4 Macro Overview 5-10
5.4.1 Hierarchical Use of the Five Macros 5-11
5.4.2 $RSUD Macro 5-12
5.4.2.1 Example 1 - FILE=TRANS 5-15
5.4.2.2 Example 2 - FILE=PRINT 5-15
5.4.3 $RSUD: Defaults Not Related to FILE 5-16
5.4.3.1 Example 1 - $RSUD Using REP 5-19
5.4.4 $RSUF Macro 5-20
5.4.4.1 Syntax of $RSUF - Defining Literals 5-24
5.4.5 $RSUL Macro 5-25
5.4.6 $RSUV Macro 5-27
5.4.7 $RSUEND Macro 5-31
5.4.8 Assembling the User CSECT 5-31
5.5 Performance Notes 5-33

Appendix A. Keywords and Fields A-1
A.1 Keyword List A-2
A.2 Keywords by Category A-5
A.2.1 Search A-5
A.2.2 Selection A-5
A.2.3 Print A-5
A.2.4 Output A-5
A.3 Keyword Definitions A-6
A.4 Fields by Category A-20
A.4.1 Data Set Fields A-20
A.4.2 Data Set Extent Fields A-20
A.4.3 Volume Fields A-20
A.4.4 Billing Fields A-21
A.4.5 Miscellaneous Fields A-21
A.4.6 Predefined Literal Fields A-21
A.4.7 VSAM Fields A-21
A.4.8 IPC Fields (CA-ASM2) A-22
A.4.9 SMS Fields A-23
A.4.10 TMC Fields (CA-1) A-24

A.4.11 VMF Fields (CA-Dynam/TLMS) A-25
A.5 Field Definitions A-26

Glossary X-1

Index X-7

Chapter 1. Introduction

RSVP (Report Selection and Variable Processing) is a command facility used with the CA-ASM2, CA-1 Version 5.2+, and CA-Dynam/TLMS Version 5.5+ storage management products. It is the most comprehensive storage reporting facility available today, and a major advance toward providing complete reporting flexibility. Reporting provides information on problems, exceptions, and the effectiveness of storage management. Although there are many standard reports, you can easily add or delete fields and create reports tailored to the specific requirements of your installation.

RSVP has the following capabilities:

- Scans entries in the VTOC, CA-ASM2 Integrated Product Catalog (IPC), CA-1 Tape Management Catalog (TMC), CA-Dynam/TLMS Volume Master File (VMF), VSAM catalogs (both ICF and non-ICF catalogs), and Storage Managed Subsystem (SMS) constructs database.
- Reports on over 100 different elements associated with a data set or VSAM cluster. VSAM entries are treated as if the DSCB contains the same information as non-VSAM data sets.
- Provides exception reporting using selection criteria.
- Uses English-like keywords and fields names for selection criteria and output report formats.
- Generates preprocessing commands to be passed on to other programs.
- Performs billing on DASD occupancy.

With RSVP you are able to report on existing volume information (based on VTOC and catalog information) using extensive selection criteria. Keywords provide the ability to select, sort, report, or create billing data based on the values of the corresponding fields in the VTOC, IPC, TMC, VMF, ICF catalog, and SMS database.

Also included are pattern masking facilities that let you scan on virtually anything within a data set name.

You can search multiple sources of primary information during a single invocation (DASD volumes, mass storage volumes, catalogs, SMS Storage Groups, TMC, VMF, and the IPC) and present that information in a single report.

Multivalued fields are supported. These are fields for which there are a variable number of entries within the same record. As an example, when you define a VSAM

cluster you may specify a number of volumes on which the data set may reside. You can report, sort, or select based on any one of these values. For example, you might request that the first five candidate volumes be reported for a VSAM cluster for which the second candidate volume was TSO002.

You can also use RSVP as a preprocessor to any number of standard facilities, such as archive, backup, and space management.

Finally, you can perform DASD billing calculations and create billing files and reports. You can break out charges by account code and apply adjustable charge rates for space usage.

Figure 1 in this chapter illustrates the logical flow.

Reporting, preprocessing, and billing facilities are discussed in separate chapters in this guide. Each chapter contains many examples of how you might use the RSVP component to manage your data storage.

1.1.1 $RSVP Command

Processing can be executed with the $RSVP command from ISPF panels, in TSO command mode, or in batch mode. The ISPF interface offers a full-screen, menu-driven approach towards using panels under ISPF. Panels provide options for volume selection, data set name pattern masking, and Boolean logic attribute comparison for record selection. You can browse output from RSVP, edit the data, or print the output being displayed. The online ISPF interface is completely documented in the *CA-ASM2 ISPF User Interface Guide*. Review the entire section before using the panels.

The $RSVP command provides criteria for selecting data sets for further processing. The command format is $RSVP, optionally followed by keywords and fields, ($RSVP keyword ...). If you do not specify keywords, a listing of data sets in the system catalog whose names begin with your user ID is provided. Keywords are grouped by reporting, transaction processing, and billing functions. You may include as many keywords as can fit on a TSO command line. Keywords may be in any order. There are over 80 keywords and over 500 fields available. For a complete list, see Appendix A, “Keywords and Fields.”

1.1.1.1 Executing the $RSVP Command in Batch Mode

The batch command processor program $ASMBMON can be used to execute $RSVP commands in batch mode. The TSO Batch Terminal Monitor Program - IKJEFT01, can also be used, however, the use of $ASMBMON is recommended for performance reasons.

RSVP users are provided the RSVPCMDU procedure to invoke the $ASMBMON program. It contains all of the JCL necessary to execute $RSVP commands. CA-ASM2 users should use the ASM2CMDU procedure to execute $RSVP commands. Examples in this guide show the use of both the RSVPCMDU and ASM2CMDU procedures.

Examples which are specific to CA-ASM2 environments use the ASM2CMDU procedure.

Note: To continue a batch $RSVP command on additional lines, you simply add a hyphen at the end of the line to be continued. For example,

```
$RSVP  VOL(*)  INDEXTOTALS(1)  TRK -
NOPRINT(NEW (DSCOUNT ALLOC USED UNUSED EXT DSNAME))
```

REPORT SELECTION AND VARIABLE PROCESSING

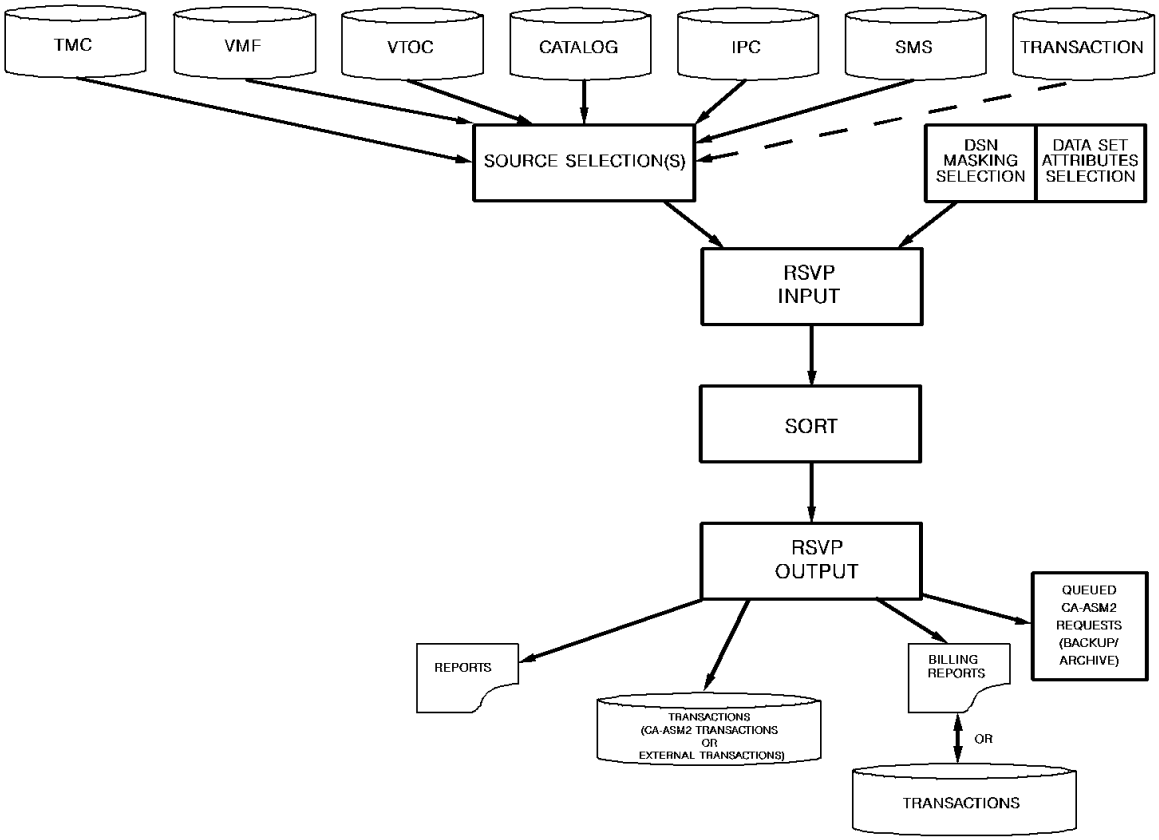


Figure 1 - RSVP Logical Flow

1.1.2 RSVP Controls

Four different points of control determine how RSVP functions in your system:

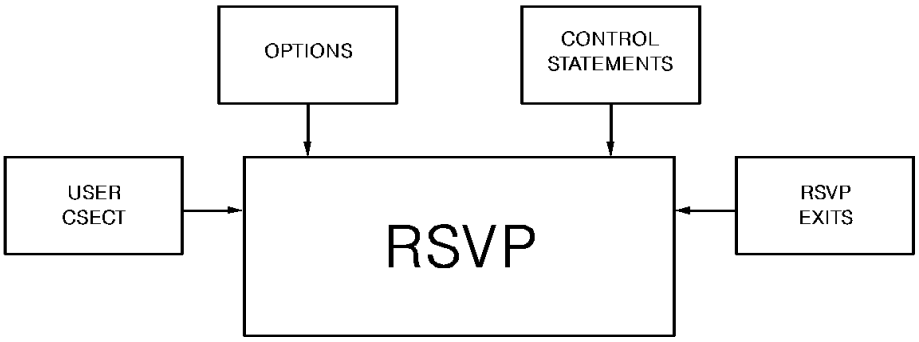


Figure 2 - RSVP Points of Control

At the time RSVP was installed, your installation specified various parameters that could affect the way RSVP operates in your system.

Part of the installation and customization process for RSVP is selecting options for your environment. Customizing (for example, changing defaults, adding more fields, or generating standard output transactions in the user CSECT) is typically done by your installation's systems programmer or system administrator. Discussions and examples in this guide are based on the standard default options. If some of the options were changed in your system, RSVP functions differently. You may want to contact your systems programmer for a list of the specific options selected for your system, and record that information here for future reference. $RSEXT1 and RSEXT2 are two standard user exits provided for controlling the $RSVP command process. A default user CSECT (user control section) is provided for tailoring. The user CSECT contains system parameters, field definitions, and output lists used by RSVP. Both the user exits and the user CSECT can be easily modified. For a detailed description of these exits, see User Exit Descriptions in Chapter 2, “Reporting.” For detailed information on modifying them, see RSVP User Exits in Chapter 5, “Tailoring.” Detailed information on the user CSECT is also provided in Chapter 5, “Tailoring.”

1.1.3 Keyword Defaults

Although the user CSECT determines the defaults for these keywords, the following defaults were established for the shipped version of RSVP:

BLIST(BILLLIST)
CHARS
DATASPACE
DDNAME($RSOUT)
INBLIST
KBYTES
LINES

LIST(SPACE)
MERGE
PRINT
SORT(DSNAME)
TOTALS
USERCSECT($RSUSER)

The defaults in your installation may be different from the ones shown. Consult your systems programmer. For information on customizing, see User CSECT in Chapter 5, "Tailoring."

1.1.4 Terminology

Term	Definition
dsname	Data set name. This can be up to 44 characters, in accordance with OS data set naming conventions.
dsnmask	Selection pattern mask for data set name (see Pattern Masking in this chapter.
high-level index	A one- to eight-character string that identifies the beginning of a dsname. The character following the string in the dsname is a period. The high-level index is also referred to as the first node.
high-level qualifier	A character string at the beginning of a dsname. It may be a full or partial index level.
low-level index	A one- to eight-character string that identifies the ending of a dsname. The character preceding the string in the dsname is a period. The low-level index is also referred to as the last node.
low-level qualifier	A character string at the end of a dsname. It may be a full or partial index level.
qualifier	A one- to eight-character string contained in a dsname. It may not begin or end with a period.
user list	A user-supplied list of report fields to be printed in a report.
volser	A six-character volume serial identifier.

Examples of indexes and qualifiers for dsname CAI.DOC.TEXT follow:

High-level index:	CAI
Low-level index:	TEXT
High-level qualifier:	CAI.DOC and CAI.DOC.TE
Low-level qualifier:	TEXT and OC.TEXT
Qualifier:	O and I.DOC.T

1.1.5 Pattern Masking of Data Set Names

The pattern masking facility allows data sets to be identified and selected based on a dsname mask or pattern. They are selected through the use of the LIKE keyword and NOTLIKE keyword (see the chapter "Reporting.") A pattern consists of a series of character strings that are one to eight characters long and are separated by periods. Each one of these character strings is called an index level or node. To establish patterns, use the asterisk sign (*) or the minus sign (-) or the not sign (¬) as symbolic replacement of characters in the string.

Asterisk

An asterisk (*) placed as the last character of an index level indicates that any character, or the absence of a character, may be substituted. Therefore, $CAI** matches the characters $CAI followed by any two characters, any single character, or nothing.

```
$CAI01.ABC*.LOAD matches
      $CAI01.ABC.LOAD
      $CAI01.ABCD.LOAD
  but not
      $CAI01.ABCDE.LOAD
      $CAI01.LOAD
```

An asterisk (*) placed anywhere else in the index level indicates a matching character, but not the absence of a character, may be substituted. For example,

```
**C matches
  ABC
  CCC
but does not match
   C
   BC
```

Minus Sign

A minus sign (-) can be used in two ways: (1) at the character level or within an index level, and (2) at the index level.

The minus sign (-) used as the last character of an index level within the pattern indicates that asterisks are to be appended to the index level to create the maximum length of eight characters.

```
$CAI01.ABC-.LOAD matches
      $CAI01.ABC.LOAD
      $CAI01.ABCD.LOAD
      $CAI01.ABCDEFGH.LOAD
  but not
      $CAI01.AB.LOAD

$CAI01.$- matches
      $CAI01.$ABC
      $CAI01.$ABCDEF
```

A minus sign can be the first character of an index level and can be followed by other characters. That index level of the target data set ends with the characters specified. You can replace the minus sign with any number of characters, including zero; however, the index level cannot exceed eight characters. For example, -LIST matches data sets with the index level of LIST, CLIST, LINKLIST.

```
XXX.-LIST matches
      XXX.LIST
      XXX.CLIST
      XXX.OUTLIST
      XXX.LINKLIST
```

A minus sign as the only character of an index level indicates that any number of index levels, including zero, may be present in the target data set name. For example, a pattern consisting of -.LOAD matches all data sets ending in LOAD regardless of the number of other index levels present.

```
$CAI01.$-.- matches
      $CAI01.$ABC
      $CAI01.$ABCDEF
      $CAI01.$JUNK.A.B.DATA

$CAI01.-.LOAD matches
      $CAI01.LOAD
      $CAI01.ABC.LOAD
      $CAI01.ABCD.LOAD
      $CAI01.A.B.C.D.E.LOAD
  but not
      $CAI01.LOAD.X

-.SPFTEMP1 matches
      USER1.SPFTEMP1
      XYZ.ABC.SPFTEMP1
```

Not Sign

A not sign (¬) used as the last character of an index level or preceding a trailing minus sign (-) indicates any character, but not the absence of a character, can be substituted. This is useful to specify the exact length of the string to be checked.

```
$CAI01.A**¬.LOAD matches
      $CAI01.ABCD.LOAD
  but does not match
      $CAI01.ABC.LOAD
      $CAI01.ABCXY.LOAD
```

If you use the not sign in any position other than the last character of a level or preceding a trailing minus sign, it has the same meaning as an asterisk (*).

```
$CAI01.A**¬-.LOAD matches
      $CAI01.ABCDEFGH.LOAD
  but not
      $CAI01.ABC.LOAD
```

Note: The not sign (¬) is an EBCDIC X'5F'. On some keyboards X'5F' is represented by a character other than ¬. If you do not have the ¬ character on your keyboard, see the EBCDIC code chart for your keyboard to determine which character generates X'5F'.

1.2 Using This Guide

This guide is organized into a basic and advanced format. Everything you need to know about the RSVP component is explained in this guide. To use the reporting, preprocessing, and billing facilities, read Chapter 2, “Reporting,” Chapter 3, “Preprocessing,” and Chapter 4, “DASD Billing.” Once you are familiar with using RSVP, you might want to know more about tailoring it to your environment. Read Chapter 5, “Tailoring” on page 5-1 for advanced information on modifying and customizing the user exits and user CSECT, and a summary of performance notes. Messages issued from RSVP are explained in the *CA-ASM2 Message Guide*.

At the back of this guide is an appendix and an index. Appendix A, “Keywords and Fields” contains a complete list and description of keywords and fields arranged in alphabetical order for fast referencing. Keywords are indexed in two ways: as a group by "Keywords" and individually by keyword name. Index entries show the page on which the keyword is defined and the pages, throughout the guide, that contain examples using the keyword.

Each chapter in this guide shows the applicable command format, corresponding ISPF panel, keyword descriptions, and examples of commands and output reports. Defaults are underlined. The example reports have been edited where necessary to fit them into a sample report format. They may look slightly different from the actual run version. A "|" indicates sections of reports that were deleted to save space.

Note: Throughout this guide RSVP refers to a component of CA-ASM2. $RSVP refers to a command that runs under the RSVP component.

1.3 Summary of New Features for Version 4.2

Each of the following Version 4.2 enhancements for CA-ASM2 are described in detail in the new set of documentation issued for this version.

1.3.1 9999 Files Tape Support

CA-ASM2 now supports up to 9,999 files on a single CA-ASM2 Archive or Backup tape.

- Allows unload to tape to hold 9999 files
- $FORMAT automatically updates LOxxx file(s)
- Converts old format LOxxx files to new format

1.3.2 SMS Support Enhancements

The Management Class fields controlling expiration may now optionally be used to control when CA-ASM2 will expire, archive and backup versions. In addition, CA-ASM2 invokes the ACS class selection routines prior to reload to handle situations in which the ACS class selection rules have changed and the data set is assigned to different classes.

- Reload pre-drives ACS routines to determine SMS classes during dynamic allocation
- Uses SMS MGTCLASS for retention values during archives and backups

1.3.3 ISPF Interface Enhancements

The CA-ASM2 ISPF application has been enhanced to provide additional functionality to end users of CA-ASM2. Online panels are provided to perform common functions such as requesting the restore of all data sets for an application through a new Application construct and restoring a volume from a specific point in time. Customization capabilities are built in through the use of a profile, which removes options that are inappropriate for general users.

- Group data sets at the application level
- High-level qualifier masking and wild cards

1.3.4 IXR Enhancements

The temporary name created by IXR to restore a data set now supports the specification of up to three alias levels for non-VSAM data sets and two levels for VSAM data sets.

Also a New RLDTMPNM PARMLIB keyword for data set high-level qualifier usage during reload UCAT processing has been added.

1.3.5 CA-ASM2 Workstation

CA-ASM2 integration with the CA-ASM2 Workstation has been enhanced to provide extensive administration, reporting, and storage management capabilities from a Windows-based graphical user interface. CA-ASM2 Workstation is a separately licensed product.

1.3.6 $DASDMNT

Support for SMS management class retention of archived data sets has been implemented via the use of a new keyword $SMSRTPD which can be coded in the SYSIN stream.

1.4 Documentation Changes

- A new document, the *ISPF User Interface Guide* has been added to the documentation set. This document describes all the new extended features developed for Version 4.2. In addition, all ISPF panels that were previously described in Appendix A of the *CA-ASM2 System Reference Guide* and Appendix A of the *CA-ASM2 RSVP User Guide* have been integrated into this document so that a single source can be referenced for all ISPF applications.
- All guides have been updated with relevant Version 4.2 information.
- The syntax throughout the documentation has been given a new look. Information has been provided to assist you in reading the syntax diagrams.
- The Troubleshooting section, found in the *CA-ASM2 System Reference Guide*, now provides information for accessing the Computer Associates home page on the Internet for additional Computer Associates products and services.

1.4.1 Removed

- *CA-ASM2 General Information Guide*. Information located in this guide has been disseminated into the *CA-ASM2 System Reference Guide* and *CA-ASM2 Getting Started*.
- Conversion guides for CA-3 and CA-Dynam/DASD.
- *CA-ASM2 Master Index*.
- Demand Analysis Request (DAR) form. You can now enter your request through StarTCC Extended Support (click on Support at www.ca.com on the Web).

1.5 Reading Syntax Diagrams

The formats of all statements and some basic language elements are illustrated using syntax diagrams. Read syntax diagrams from left to right and top to bottom.

The following terminology, symbols, and concepts are used in syntax diagrams.

Keywords: Appear in uppercase letters, for example, COMMAND or PARM. These words must be entered exactly as shown.

Variables: Appear in italicized lowercase letters, for example, *variable*.

Required Keywords and Variables: Appear on a main line.

Optional Keywords and Variables: Appear below a main line.

Default Keywords and Variables: Appear above a main line.

Double Arrowheads Pointing to the Right: Indicate the beginning of a statement.

Double Arrowheads Pointing to Each Other: Indicate the end of a statement.

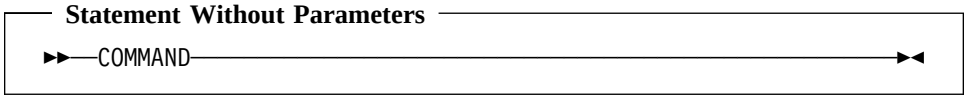
Single Arrowheads Pointing to the Right: Indicate a portion of a statement, or that the statement continues in another diagram.

Punctuation Marks or Arithmetic Symbols: If punctuation marks or arithmetic symbols are shown with a keyword or variable, they must be entered as part of the statement or command. Punctuation marks and arithmetic symbols can include:

,	comma	>	greater than symbol
.	period	<	less than symbol
(	open parenthesis	=	equal sign
)	close parenthesis	¬	not sign
+	addition	–	subtraction
*	multiplication	/	division

The following is an example of a statement without parameters.

Statement Without Parameters

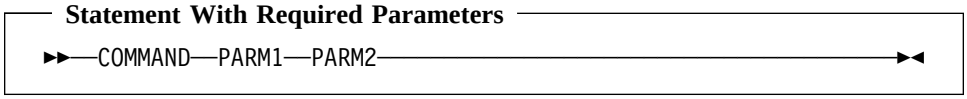


You must write:

```
COMMAND
```

Required parameters appear on the same horizontal line (the main path of the diagram) as the command or statement. The parameters must be separated by one or more blanks.

Statement With Required Parameters

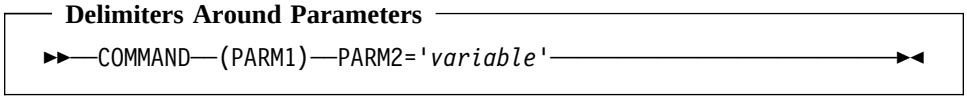


You must write:

```
COMMAND PARM1 PARM2
```

Delimiters such as parentheses around parameters or clauses must be included.

Delimiters Around Parameters

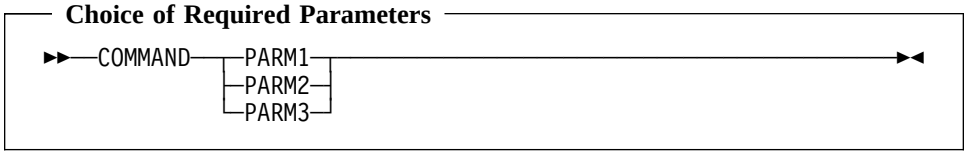


If the word “variable” is a valid entry, you must write:

```
COMMAND (PARM1) PARM2='variable'
```

Where you see a vertical list of parameters as shown in the following example, you must choose one of the parameters. This indicates that one entry is required and only one of the displayed parameters is allowed in the statement.

Choice of Required Parameters



You can choose one of the parameters from the vertical list, such as in the following examples:

```
COMMAND PARM1
COMMAND PARM2
COMMAND PARM3
```

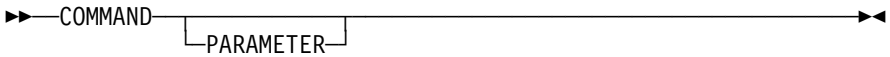
A single optional parameter appears below the horizontal line that marks the main path. When a required parameter in a syntax diagram has a default value, it indicates the value for the parameter if the command is not specified. If you specify the command, you must code the parameter and specify one of the displayed values.

Default Value for a Required Parameter

If you specify the command, you must write one of the following:

```
COMMAND PARM1=NO PARM2
COMMAND PARM1=YES PARM2
```

Optional Parameter

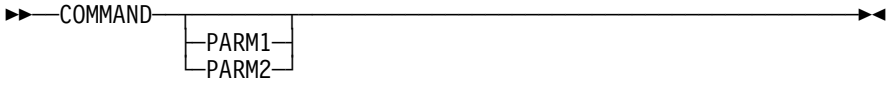


You can choose (or not) to use the optional parameter, as shown in the following examples:

```
COMMAND
COMMAND PARAMETER
```

If you have a choice of more than one optional parameter, the parameters appear in a vertical list below the main path.

Choice of Optional Parameters

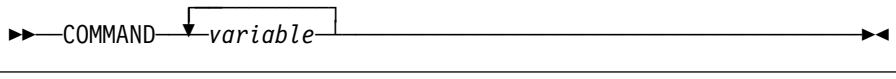


You can choose any of the parameters from the vertical list, or you can write the statement without an optional parameter, such as in the following examples:

```
COMMAND
COMMAND PARM1
COMMAND PARM2
```

For some statements, you can specify a single parameter more than once. A repeat symbol (a backward-pointing arrow above the main horizontal line) indicates that you can specify multiple parameters. The following examples include the repeat symbol.

Repeatable Variable Parameter

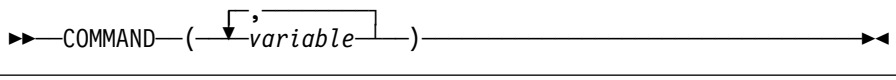


In the above example, the word “variable” is in lowercase italics, indicating that it is a value you supply, but it is also on the main path, which means that you are required to specify at least one entry. The repeat symbol indicates that you can specify a parameter more than once. Assume that you have three values named VALUEX, VALUEY, and VALUEZ for the variable. Some of your choices are:

```
COMMAND VALUEX
COMMAND VALUEX VALUEY
COMMAND VALUEX VALUEX VALUEZ
```

If the repeat symbol contains punctuation such as a comma, you must separate multiple parameters with the punctuation. The following example includes the repeat symbol, a comma, and parentheses.

Separator with Repeatable Variable and Delimiter

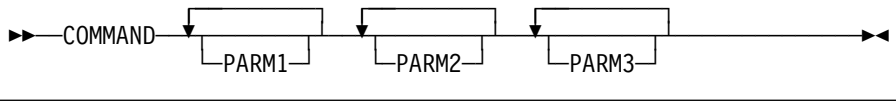


In the above example, the word “variable” is in lowercase italics, indicating that it is a value you supply. It is also on the main path, which means that you must specify at least one entry. The repeat symbol indicates that you can specify more than one variable and that you must separate the entries with commas. The parentheses indicate that the group of entries must be enclosed within parentheses. Assume that you have three values named VALUEA, VALUEB, and VALUEC for the variable. Some of your choices are:

```
COMMAND (VALUEC)
COMMAND (VALUEB,VALUEC)
COMMAND (VALUEB,VALUEA)
COMMAND (VALUEA,VALUEB,VALUEC)
```

The following example shows a list of parameters with the repeat symbol.

Optional Repeatable Parameters



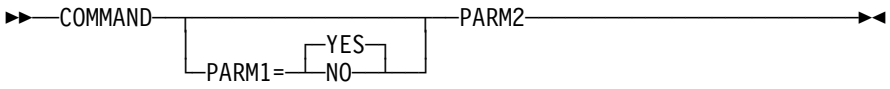
Some choices you can make include:

```
COMMAND PARM1
COMMAND PARM1 PARM2 PARM3
```

COMMAND PARM1 PARM1 PARM3

When a parameter in a syntax diagram is above the line, for example, YES in the following diagram, its special treatment indicates it is the default value for the parameter. If you do not include the parameter when you write the statement, the result is the same as if you had actually specified the parameter with the default value.

Default Value for a Parameter



Because YES is the default in the example above, if you write:

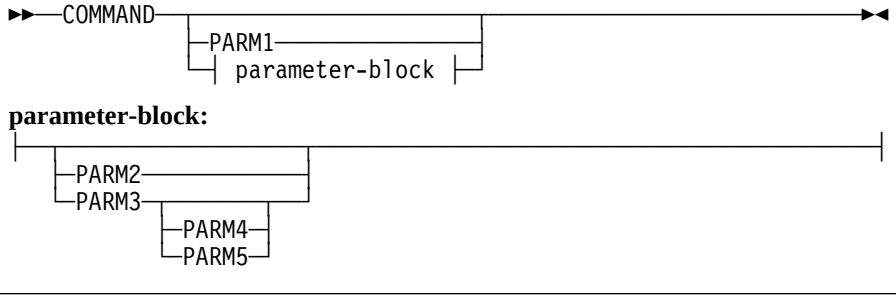
COMMAND PARM2

you have written the equivalent of:

COMMAND PARM1=YES PARM2

In some syntax diagrams, a set of several parameters is represented by a single reference, as in this example:

Variables Representing Several Parameters



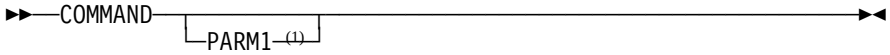
The “parameter-block” can be displayed in a separate syntax diagram.

Choices you can make from this syntax diagram therefore include (but are not limited to) the following:

COMMAND PARM1
COMMAND PARM3
COMMAND PARM3 PARM4

Note: Before you can specify PARM4 or PARM5 in this command, you must specify PARM3.

A note in a syntax diagram is similar to a footnote except that the note appears at the bottom of the diagram box.



Note:
1 This is a note about the item.

Chapter 2. Reporting

RSVP lets you design individualized reports customized for your installation's specific needs. You control the appearance of the report, its contents, sequence, sort criteria, headings, and totals. There are over 20 default report formats (see LIST on page 2-24). Each output format lists relevant field names. You can modify the default formats by simply adding, deleting, or replacing specified fields, or create a new report format.

There are three distinct report generating phases:

- **Input Selection/Filtering**
- **Sort**
- **Output Format**

The Input Selection/Filtering phase lets you choose data to obtain only the desired records. You define selection criteria by specifying input sources and data set name or attribute information. You can obtain information from the VTOC, system catalogs, the CA-ASM2 Integrated Product Catalog (IPC), the CA-1 Tape Management Catalog (TMC), the CA-Dynam/TLMS Volume Master File (VMF) and the Storage Management Subsystem (SMS) construct database and combine this into one report. This allows you to receive information in a single report about all versions of a data set that resides on both disk and tape (Backup, Archive, and Full-volume Backup).

The Sort phase lets you determine the sort order of the fields for the selected data. You can sort on up to eight fields in ascending and descending sequence and divide the output into sections based on the primary input sources, or treat the data as one big source.

The Output Format phase lets you select the report format, produce the report, and generate a transaction file for later processing. You do not have to reenter repetitive report formats each time you invoke the $RSVP command. You simply define the report format and assign it a unique report name that is referenced in later command execution.

Although the extensive selection criteria and reporting capabilities virtually eliminate the need for user exits, two user exits, $RSEXT1 and $RSEXT2, are provided that let you easily tailor the $RSVP command process. For more information, see User Exit Descriptions on page 2-35.

2.1 Input Selection/Filtering Phase

Selection criteria allows you to create a report containing the exact data set information needed. You control the input selection process and record filtering in two ways:

1. **Source Selection -** Limiting the search to the specified input source:
   - DASD volumes
   - IPC
   - System catalogs
   - SMS database
   - Transaction data
   - TMC (CA-1)
   - VMF (CA-Dynam/TLMS)

   Note: RSVP can search more than one primary input source in a single pass, except transaction data. If transaction data is supplied, only it is processed.
2. **Data Set Selection -** Selecting data sets by comparing against levels of the dsnames, or by identifying attributes of the data set.

An overview of the Input Selection/Filtering phase is shown in RSVP Logical Flow on page 1-3.

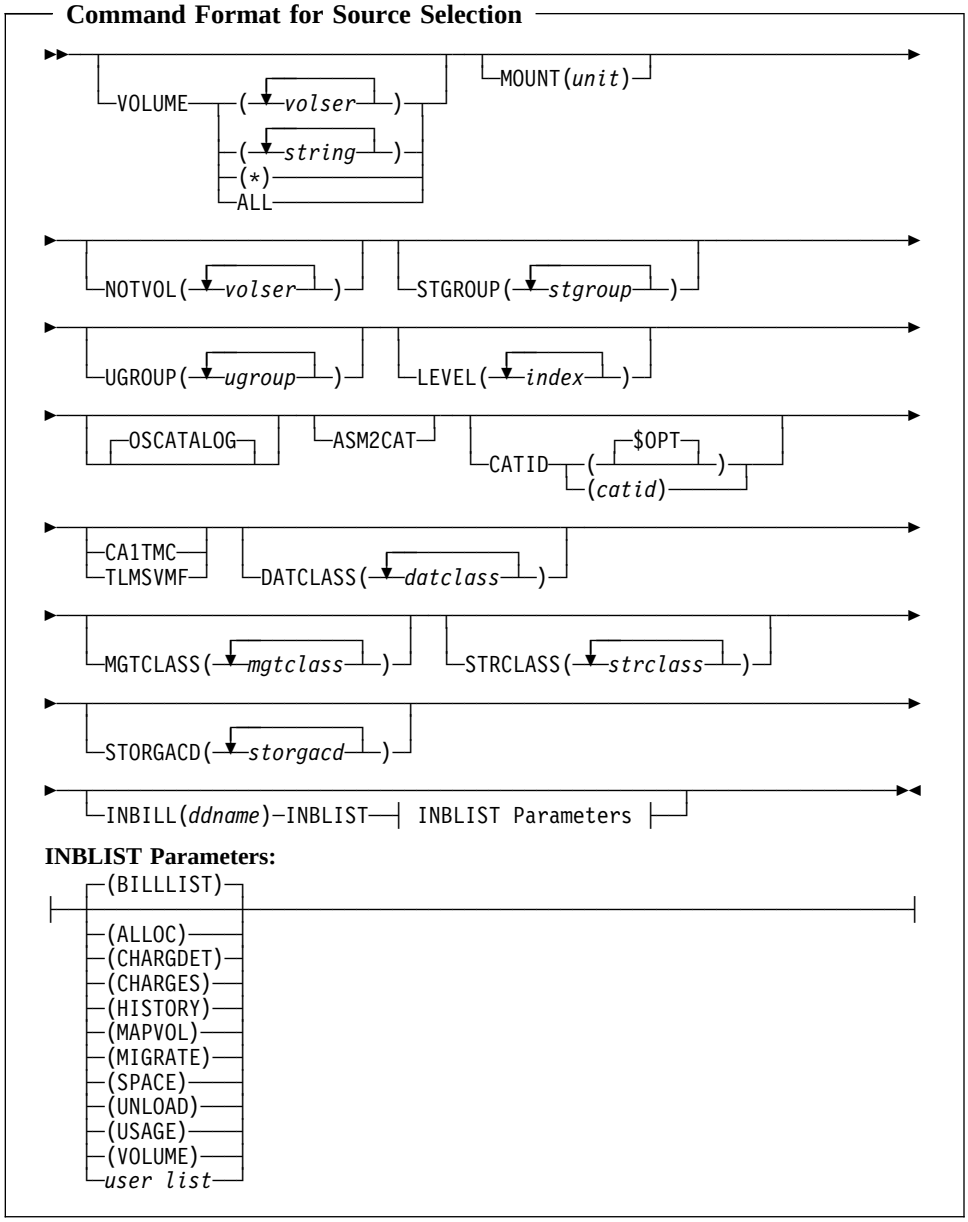
2.1.1 Source Selection

This section provides the following information related to source selection:

- Command format
- Keyword descriptions
- ISPF for input source selection

2.1.1.1 Command Format for Source Selection

Command Format for Source Selection



INBLIST Parameters:

2.1.1.2 Keyword Descriptions for DASD Volumes

RSVP can search any number of online volumes.

VOLUME Identifies the following volumes to search:

1. Specific volumes by the full six-character volume serial number, such as VOLUME(SYS001 SYS005 WORK01 WORK02).
2. Generic volumes beginning with a 1- to 6-character string, such as VOLUME(SYS WORK).
3. All volumes, as referenced by an asterisk (*), such as VOLUME(*).
4. All volumes, as referenced by the keyword ALL, such as VOLUME(ALL).

Note: VOLUME and NOTVOL can be used together to specify which volumes are to be searched. For example, `VOL(SJ) NOTVOL(SJ0005)` searches for all SJ volumes except SJ0005.

MOUNT Specifies a unit name or device type on which to mount only specific volumes. For example: `MOUNT(3390)`, `MOUNT(26C)`, or `MOUNT(SYSMNT)`. (This works online only if the MOUNT attribute is specified in SYS1.UADS.)

NOTVOL Identifies the generic or specific volume serial numbers to be excluded from selection. NOTVOL overrides VOLUME.

STGROUP Identifies the volumes in the specified SMS Storage Groups that are searched. Storage Group names may be specified with up to 30 characters. Pattern masking is supported for STGROUP.

UGROUP Specifies the non-SMS esoteric unit group name(s) that is searched. All volumes in this unit group or groups are scanned.

2.1.1.3 Keyword Descriptions for IPC and System Catalogs

RSVP can search the IPC, and the system catalog on high-level indexes. If you do **not** specify the VOLUME keyword or INBILL keyword, or if you **do** specify the OSCATALOG keyword, the default is to search the system catalog on your user logon ID or batch job prefix (job name minus the last character). The LEVEL keyword, when specified, overrides this default and searches other high-level indexes.

LEVEL Identifies specific nodes to be used in the system catalog or IPC searches, such as LEVEL($CAI01 $CAI03). For other primary input sources, LEVEL is used to limit the DSNs returned.

If RSVP searches the system catalog or IPC in batch without the keyword LEVEL, the default user ID is the job name minus 1 character, (for example, job $CAI01A defaults to a node of $CAI01).

If RSVP searches the catalog or IPC under TSO, the user index is the default level searched. For IPC searches, LEVEL(*) indicates the entire IPC should be searched; other criteria determines the record filtering and selection.

Note: Specifying LEVEL(*) for IPC searches causes the entire IPC to be read. Avoid doing this unless it is absolutely necessary or include the IPCTYPE and/or IPCUTYPE keywords to limit the records processed.

LEVEL can be used in conjunction with other keywords such as LIKE and BEGIN.

The keywords CLUSTER, DATA, and DATASPACE used with LEVEL( ) to select specified high-level indexes are ignored unless the VOL( ) keyword is also used.

OSCATALOG Specifies that catalog searches are to be performed on the nodes specified by the LEVEL keyword. This option lets you search ICF catalogs and other primary sources with the same invocation.

ASM2CAT Specifies the ASM2 IPC catalog to be used as a primary source for input. The IPC is searched on the nodes specified by the LEVEL keyword. If no catalog identifier (CATID) is specified, the catid $OPT is used. $OPT specifies that fields in the $OPTIONS module be used to define which IPC is to be searched. $OPTIONS is discussed in the *CA-ASM2 Planning Guide*.

Note: $DEFRAG volume records are queried from the IPC if, and only if, a $DEFRAG field is requested. Otherwise, only unload records are returned. Also, when the IPCVTYPE field is specified in a list or on the command line, RSVP returns IPC VSAM path records and IPC unload data set records. For path records, IPCVTYPE contains the value R. The DSNAME field is the name of the path. ASSOCC and ASSOCG are also valid for path records. The value of any other non-IPC type fields are unpredictable for path records.

Queries that do not access the IPCVTYPE field only search IPC unload records.

For examples showing the use of ASM2CAT, see 2-55.

CATID(catid) Specifies which CA-ASM2 IPC is to be searched. If no catalog identifier (catid) is specified, the catid $OPT is used. $OPT specifies that fields in the $OPTIONS module be used to define which IPC is to be searched. The $VPREFIX field of $OPTIONS is used for the high-level qualifier of an IPC data set name. $OPTIONS is discussed in the *CA-ASM2 Planning Guide*.

Note: CATID is usually only needed when the IPC is the primary input source. However, CATID may also be required when the field LASTBKP is specified and volumes or system catalogs are the primary input source.

2.1.1.4 Keyword Descriptions for SMS

RSVP can search the SMS database for data sets associated with SMS Data, Management, and Storage Classes and SMS Storage Groups. You can specify field names associated with each SMS Class or Group name to obtain specific attribute information (see SMS examples beginning on 2-46).

DATCLASS Indicates that SMS Data Classes are to be searched.

MGTCLASS Indicates that SMS Management Classes are to be searched.

STORGACD Indicates that SMS Storage Groups are to be searched.

STRCLASS Indicates that SMS Storage Classes are to be searched.

2.1.1.5 Keyword Descriptions for Transaction Data

RSVP can search data that exists in sequential files that have been predefined to RSVP as stored report field lists. For more information on these lists, see LIST on page 2-24.

INBILL Defines the DDNAME for a sequential file being input into RSVP.

INBLIST Specifies the name of the field list that defines the records contained in the sequential file referenced in INBILL.

2.1.1.6 Keyword Descriptions for CA-1 TMC and CA-Dynam/TLMS VMF

RSVP can search the TMC and VMF on almost all fields defined in the TMC and VMF (see Appendix A for complete listing). By default, every record is processed. However, records can be limited by specifying the IF and AND keywords. In addition, if a search by data set name is required, pattern masking and the use of other data set name selection keywords, such as BEGIN and ENDING can be used.

CA1TMC Specifies that the CA-1 Tape Management Catalog is to be used as the primary input source. When this keyword is specified, the TMC must be allocated with the DDNAME of TAPEDB prior to the issuing of the command.

TLMSVMF Specifies that the TLMS Volume Master File is to be used as the primary input source. When this keyword is specified, the VMF must be allocated with the DDNAME of CAIVMF prior to the issuing of the command.

Note: CA1TMC and TLMSVMF are mutually exclusive. They cannot be specified on the same $RSVP command.

2.1.1.7 ISPF for Input Source Selection

The RSVP Primary Input Source(s) Selection panel displays when you enter I on the OPTION line on the RSVP Primary Selection Menu, or on the COMMAND line of any RSVP panel.

This panel lets you search the ASM2 IPC, system catalogs, specific generic volume serials, and the SMS database. See Source Selection on page 2-2 for a description of the keywords on this screen.

RSVP Primary Input Source(s) Selection

```
------------------ RSVP Primary Input Source(s) Selection ------------------
COMMAND ===>
Search volumes ===>
               ===>
Do NOT search  ===>
 these volumes ===>
MSVGPs         ===>
Esoterics      ===>
Volumes in SMS  ==>
Storage Groups  ==>
SMS Data Classes =>
SMS Mgmt Classes =>
SMS Stor Classes =>
SMS Stor Groups  =>

 Enter any non-blank character to activate the following primary sources:
      ( X ) Search OS Catalogs on Index(es)
      (   ) Search ASM2 Integrated Product Catalog on Index(es)
      (   ) Search CA-1 Tape Management Catalog (TMC)
      (   ) Search TLMS Volume Master File (VMF)
ASM2 Catlg ID  ===>          (Optional ASM2 catalog identifier)
Mount Unit     ===>          (unit name for volume mounts)
```

2.1.2 Data Set Selection

This section provides the following information related to data set selection:

- Data set name selection
- Command format
- Keyword descriptions
- ISPF
- Attribute selection
  - Command format
  - Keyword description
  - ISPF

2.1.2.1 Data Set Name Selection

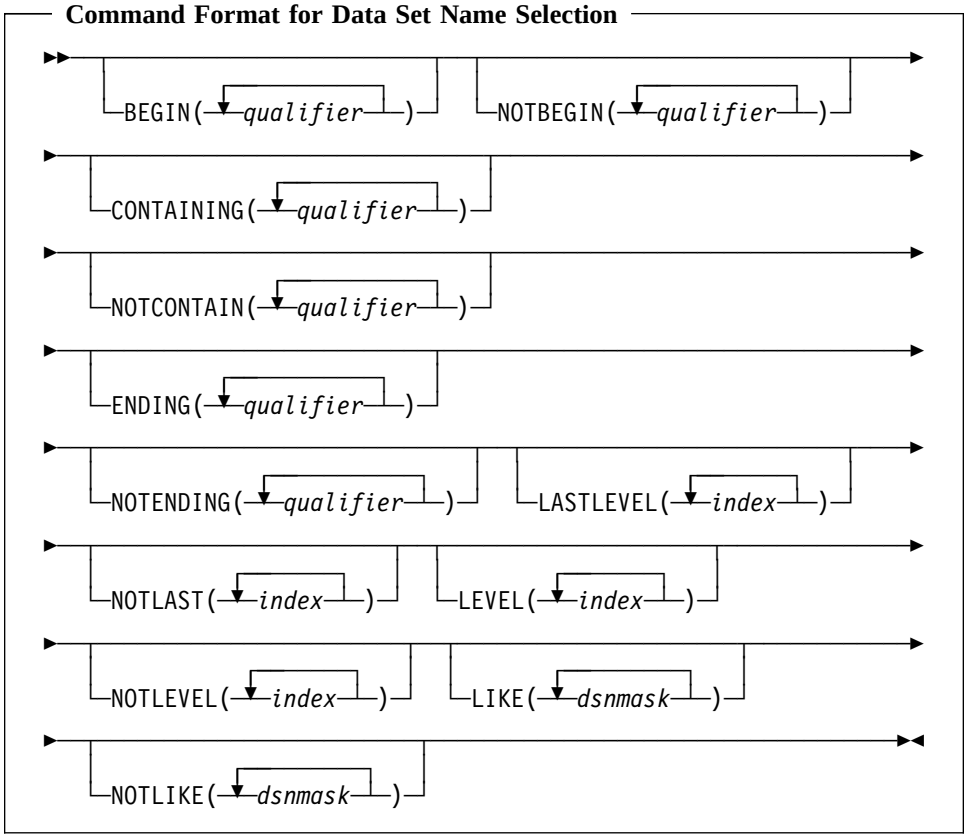
Selection based on character values in the data set name lets you generate reports for small groups of data sets. You can search on data set pattern names using a versatile pattern masking technique. Data sets are selected through the use of the LIKE and NOTLIKE keywords. A pattern consists of a series of character strings that are one to eight characters long and are separated by periods. Each one of these character strings is called an index level or node. To establish patterns, you use the asterisk sign (*), the minus sign (-), or the not sign (¬) as symbolic replacement of characters in the string. For a complete description of pattern masking see Pattern Masking on page 1-6.

Numerous keywords help control the data sets being selected by comparing against portions of the data set name. Each of the keywords can have multiple operands separated by a blank or a comma. These keywords include index-level comparisons, beginning and ending character strings in the data set name, and any specified character string. There is also an inverse keyword that excludes data sets on the same basis.

Only one statement with keywords such as LIKE/NOTLIKE is processed. If more than one is specified, the last statement entered is processed. It is acceptable to specify more than one data set per keyword statement.

Note: Keyword operands, such as data set names, must begin with an alphabetic or special character, not a numeric and they cannot end with a period.

2.1.2.2 Command Format for Data Set Name Selection



Keyword Descriptions for Data Set Name Selection:

For examples showing the use of these keywords, see Using Data Set Name Selection Keywords on page 2-42 and Using Data Set Name Simple Pattern Masking on page 2-44.

BEGIN Specifies the high-level qualifiers with which data set names must begin. To be selected, a data set name must be at least one character longer than the qualifier. For example:

```
BEGIN(CAI SYS1.A)
```

NOTBEGIN Excludes data sets with the specified high-level qualifier(s). For example:

```
NOTBEGIN(CAI SYS1.A)
```

CONTAINING Specifies qualifiers that can be located anywhere within the data set name and that cause the data set to be selected. For example:

```
CONTAINING(MAC ARCH.$)
```

NOTCONTAIN Excludes data set names containing the specified qualifiers. For example:

```
NOTCONTAIN(MAC ARCH.$)
```

ENDING Specifies the character strings with which data set names must end. To be selected a data set name must end with the string specified. For example:

```
ENDING(LIST A.DATA)
```

NOTENDING Excludes data sets ending with the specified low-level qualifiers. For example: NOTENDING(LIST A.DATA)

LASTLEVEL Selects data sets with the specified low-level indexes. For example: LASTLEVEL(ASM CNTL $RS.ASM)

NOTLAST Excludes data sets with the specified low-level indexes. For example: NOTLAST(ASM CNTL $RS.ASM)

LEVEL Selects data sets with the specified high-level indexes. For example: LEVEL($CAI03 $CAI01.$RS)

NOTLEVEL Excludes data sets with the specified high-level indexes. For example: NOTLEVEL($CAI03 $CAI01.$RS)

LIKE Specifies the data set name pattern masks on which data sets are selected. For example: LIKE($CAI**.-.$RS.- $CAI01.-.ASM) This selects data set names matching either of these two patterns. See Pattern Masking on page 1-6 for more information.

NOTLIKE Specifies the data set name pattern masks that exclude any data sets matching the patterns. For example: NOTLIKE(CAI**.-.$RS.-$CAI01.-.ASM).. See Pattern Masking on page 1-6 for more information.

Note: For volume searches, BEGIN and LEVEL are the most efficient and provide the best performance. LIKE is the least efficient. Whenever possible, use BEGIN and LEVEL with the other data set selection keywords (CONTAIN, ENDING, and so forth) instead of LIKE. For example, specify: BEGIN(MURTO01) ENDING(DATA) instead of LIKE(MURTO01.-DATA).

To avoid having to reenter the keywords BEGIN, LEVEL, or LIKE constantly, you can use the $RSEXT1I DD statement to accomplish the same thing. See Sample Input File for EXIT1 on page 2-37 for details.

2.1.2.3 ISPF for Data Set Selection

The Data Set Selection panel displays when you enter N on the OPTION line on the RSVP Primary Selection Menu, or on the COMMAND line on any RSVP panel.

This panel lets you specify keywords that control the selection of data sets by comparing against levels of data set names, or data set pattern masks. See Keyword Descriptions on page 2-10 for a description of the keywords.

```
------------------------  RSVP DATA SET SELECTION  -------------------------
COMMAND ===>

LIKE       ===>
NOT LIKE   ===>

CONTAINING ===> DATA
NOTCONTAIN ===>

BEGIN      ===>
NOT BEGIN  ===>

ENDING     ===>
NOT ENDING ===>

LEVEL      ===>
NOT LEVEL  ===>

LAST LEVEL ===>
NOT LAST   ===>
```

2.1.2.4 Attribute Selection

RSVP can select on any element in the Format-1 DSCB. It can select on criteria such as data set size, organization, last use date, or combinations based on Boolean comparisons. The keywords IF, ANDx, ORx, ANDIFx, and ORIFx form logical expressions. Logical expressions can be expanded into compound logical expressions by adding more comparisons indicators. The x in the logical set represents an expression extender. The x is replaced with a numeric value 1-10 in the sequence of occurrence in the compound logical expression.

The format of the operands associated with these keywords is: *keyword(field operator value)* where *field* is any field defined, *value* is the value being compared against that field, and *operator* is any one of six Boolean operators:

EQ is equal to

NE is not equal to

LE is less than or equal to

LT is less than

GE is greater than or equal to

GT is greater than

Integer type fields can be compared against integer values (for example, character string 0 or 225 or 23). They compare based on value, unlike regular character string compares. So, 2 and 23 are not equal, where character strings A and ABC may be equal. When comparing an integersfield, the value in the Boolean expression (IF, OR1, ...) may be specified as an integer, NOLIMIT, or MISSING. NOLIMIT is the highest integer possible. Every integer compares less than NOLIMIT. However, some SMS integers are defined as NOLIMIT. So, to see all SMS data classes with a key length

that is not NOLIMIT, you would use DATKLEN LT NOLIMIT (less than NOLIMIT). MISSING distinguishes between an integer value that is zero (0) and an integer value that is not there. For example, MISSING distinguishes the difference between an SMS data set with a Data Class and a Data Class field that is zero versus a data set that does not have a Data Class (that field is MISSING).

The *value* specified cannot contain a delimiter (space, comma, or semicolon). To include a delimiter (namely a space) as part of the value, it must be enclosed within single quotes. This string cannot exceed the maximum length for the *field*. The following example allows four characters, a space followed by VMS, to be used as selection criteria:

```
IF(TOUTLOC EQ ' VMS')
```

Evaluation

The evaluation of the operators is a two-level hierarchy. First the value of the logical set is determined. Then the new overall value is determined by evaluating the value of this logical set with the previous overall value. The conditions within a logical set are strictly evaluated in a left to right manner, and the overall value is also determined by evaluating the individual logical sets from left to right. The keywords IF, ANDIFx, and ORIFx start a logical set. The logical set is terminated by the occurrence of the next ANDIFx, ORIFx, or is the last logical set.

Limiting Dates

Notice the method of testing dates in the following example. The ability to select data sets using a series of tests makes RSVP ideal for exception reporting. In the example, the asterisk (*) implies that the current run date, and a number of days can be subtracted from today. You do not have to enter an actual date. However, if you know the actual date, you can enter it in either Gregorian or Julian format, regardless of the format of the date field you are testing (use LSTUS for date last used in Gregorian MMDDYYYY, and LDATE in Julian YYYYDDD). Note that LASTBKP cannot be used with the test described for limiting dates (*-n).

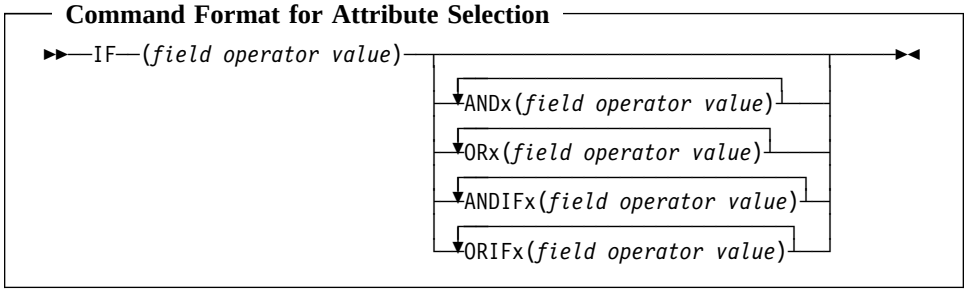
Example

```
IF(DSORG  EQ  PO)  AND1(PCT  GT  80)  AND2(ALLOC  GT  500) -
ORIF3(DSORG  EQ  PS)  OR4(DSORG  EQ  VS) -
ANDIF5(LSTUS  LT   *-30) -
ORIF6(DSORG  LT  A)
```

This command selects three possible groups of data sets:

- Any PDS with more than 80 percent used space and an allocation greater than 500 that was not used in the last 30 days. (IF, AND1, AND2, ANDIF5)
- Any sequential data set or VSAM data set not used in the last 30 days. (ORIF3, OR4, ANDIF5)
- Any data set that does not have a data set organization (one that was allocated but never opened). (ORIF6)

2.1.2.5 Command Format for Attribute Selection



Keyword Descriptions for Attribute Selection:

IF Specifies the first test or beginning of the logical expression for data set selection. It must be present for any ANDx, ORx, ANDIFx, or ORIFx to be interpreted. For example:

```
IF(DSORG EQ PO)
```

states that if the data set organization is partitioned, then select the data set for further processing. IF begins the first logical set of conditions. A logical set of conditions is comprised of an IF, ANDIFx, or ORIFx operator followed by a number of ANDx and/or ORx operators until the next ANDIFx or ORIFx is encountered. The conditions of a logical set are evaluated to true or false. The evaluation within a logical set is strictly left to right.

ANDx A maximum of ten expression extenders (ANDx,ORx,ANDIFx,ORIFx...) can follow the IF statement. The results of these expression extenders allow a data set to be selected only if all prior tests within the current logical set are also true.

ORx A maximum of ten expression extenders (ANDx,ORx,ANDIFx,ORIFx,...) can follow the IF statement. The results of these expression extenders allow a data set to be selected

regardless of all prior tests within the current logical set. For example:

```
IF(DSORG EQ PO)  OR2(DSORG EQ PS)  AND1(RECFM EQ FB)
```

selects: (1) partitioned data sets that are fixed blocked, and (2) sequential data sets that are fixed blocked.

ANDIFx A maximum of ten expression extenders (ANDx,ORx,ANDIFx,ORIFx,...) can follow the IF statement. ANDIF is a compound operator used to begin a new logical set. The condition specified on the ANDIF and any subsequent ANDx or ORx keywords that follow form the value of this logical set. For a data set to be selected, the value of this logical set and the evaluation of all prior sets must be true.

ORIFx A maximum of ten expression extenders (ANDx,ORx,ANDIFx,ORIFx,...) can follow the IF statement. ORIF is a compound operator used to begin a new logical set. The condition specified on the ORIF and any subsequent ANDx or ORx keywords that follow form the value of this logical set. A data set is selected if the value of this logical set is true regardless of the value of any previous logical sets.

Note: The total number of AND, OR, ANDIF, and ORIF keywords that you can use is ten. At each level (1-10), these keywords are mutually exclusive. This means that if you specify OR1, you cannot specify AND1, ANDIF1, or ORIF1.

2.1.2.6 ISPF for Attribute Selection

The Boolean Selection panel displays when you enter **A** on the OPTION line on the RSVP Primary Selection Menu, or on the COMMAND line on any RSVP panel.

This panel lets you control the selection of data sets by identifying data set characteristics or attributes. The 24 attribute panels let you define multiple logical sets of data set characteristics. Logical Sets are related to each other by Boolean OR operations. Attributes within the same Logical Set are related to each other by Boolean AND operations.

RSVP Boolean Selection

```
-------------------------  RSVP BOOLEAN SELECTION  -------------------------
COMMAND ===>
   DEL - Delete this Set                    GP - Goto Panel 'nn' in this Set
   NS - Next Logical Set                    NP - Next Panel in this Set
   PS - Prior Logical Set                   PP - Prior Panel in this Set
   IS - Insert a new Logical Set            XF - Expand value Field

   Logical set 01 of 01                     Panel 01 of 24 in this set

FIELD      OPERATOR   VALUE            FIELD      OPERATOR   VALUE

DSORG    ===> EQ ===> VS              ALLOC    ===>    ===>
LMDATE   ===>    ===>                 CAT      ===>    ===>
LMJOB    ===>    ===>                 CB       ===>    ===>
LMTIM    ===>    ===>                 CDATE    ===>    ===>
MDATE    ===>    ===>                 CREDT    ===>    ===>
MODID    ===>    ===>                 DSIND    ===>    ===>
LDATE    ===>    ===>                 EDATE    ===>    ===>
LRECL    ===>    ===>                 EXPDT    ===>    ===>
LSTUS    ===>    ===>                 RETPD    ===>    ===>
BLKSZ    ===>    ===>                 KBALLOC  ===>    ===>
```

Each attribute panel within a Logical Set lets you select from a list of identifying data set attribute fields. To use an attribute FIELD, you would enter a logical OPERATOR in the same row as the FIELD you wish to activate. See Attribute Selection on page 2-12 for more information, and Field Definitions on page A-26 for an alphabetical list of valid data set attribute VALUES for each field.

Notice the positional controls (DEL, NS, PS, and so on) at the top of the panel. Use these controls instead of PF keys to operate this and the other attribute panels in the set.

2.2 Sort Phase

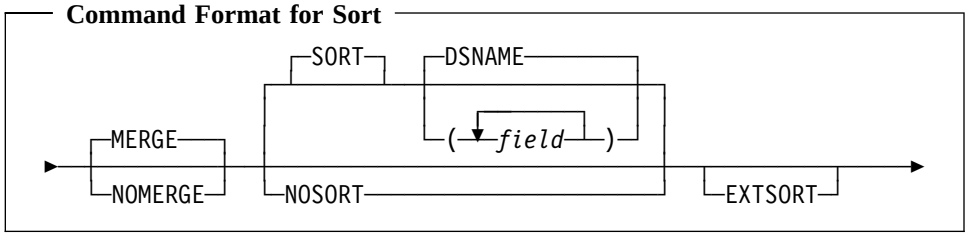
The Sort phase orders the fields for the report. You define the sort sequence by specifying the fields to be sorted. The Sort phase is an internal hashing routine that does an in-memory sort providing excellent performance.

By default, the sorted output is arranged in ascending order. The default sort parameter is SORT(DSNAME). You can indicate that any individual field is to be sorted on in descending order by placing a minus sign in parentheses after the field like this, SORT(DSN VOL DSORG(-)). For example, you could select all VSAM data sets with more than 20 CASPLITS and report them sorted descending by CASPLIT and ascending by data set name. This would place the worst offenders at the top of the list. For an example, see Example 2, Using Descending Sort on page 2-79.

To sort on multivalued fields, simply place the number of the value you want to sort on in parentheses after the field. For example, HIALLRBA(2) sorts on the second value of HIALLRBA. To sort in descending order, place a minus sign before the value like this, HIALLRBA(-2).

The MERGE keyword lets you combine the output from multiple primary input sources into one report. For example, you can request all data sets from all DASD volumes and all archived data sets from the DMC to be reported in a single report sorted by data set name. This would provide a report of all primary copies of all data sets, whether they are online or have been archived. Conversely, NOMERGE divides the output into sections based on the primary input source, and each source is sorted separately.

2.2.1 Command Format for Sort



Keyword Descriptions for Sort:

For examples showing the use of these keywords, see Sort Examples on page 2-78.

MERGE Causes RSVP to combine the output from all primary input sources into one report. MERGE is the default. For an example, see Searching More Than One Primary Input Source on page 2-74. MERGE may require a large amount of virtual storage to be acquired.

NOMERGE Causes the output to be divided into sections based on the primary input sources. Each input source is sorted separately.

SORT Determines the sort order. If no sort keyword is specified, the default is SORT(DSNAME). By default, sort is in ascending order. For example, SORT(VOLUME DSNAME) sorts first on VOLUME, then on DSNAME. To sort in descending order, place a minus sign in parentheses after the selected field. For example, SORT (DSNAME DSORG(-)).

To sort on multivalued fields, place the number of the value you want to sort on in parentheses after the field. For example: HIALLRBA(4) sorts (in ascending order) on the fourth value for the HIALLRBA field. To indicate a descending sort for a multivalued field, put a minus before the indicated value. For example: HIALLRBA(-4) sorts on the fourth value in descending order.

Keywords affecting printing such as BREAK and SUBTOTALS perform their functions based on the first sort field. The INDEXTOTALS keyword (subtotal on index levels) functions only if DSNAME is the first sort field. These keywords are described in Output Format Phase on page 2-20.

If sorting is by VOLUME, you can enhance performance by specifying CUU as the first sort field. In this case, sorting is done only for data sets on a single volume at a time. The output is in order by CUU address rather than by VOLSER.

Note: If NOPRINT is specified, you must specify SORT if using INDEXTOTALS or SUBTOTALS. NOPRINT causes a default of NOSORT.

NOSORT Bypasses the sort phase and allows printing to take place as a data set is selected. This may speed processing.

If you specify NOSORT, the report order is unit address and volume. This also nullifies the use of BREAK, SUBTOTALS, and INDEXTOTALS. These keywords are described in Output Format Phase on page 2-20.

Note: For a catalog type search, you can specify NOSORT if the level parameters are specified in ascending order.

EXTSORT Valid only if INBILL is specified. This implies the data was presorted; input was sorted in the order specified in the SORT keyword by a previous step.

This avoids an in-core sort, but allows other keywords such as BREAK, SUBTOTALS, or INDEXTOTALS to function. These keywords are described in Output Format Phase on page 2-20. This is useful and sometimes necessary when large amounts of data are involved, such as a billing history accumulation run.

For an example using EXTSORT, see Using EXTSORT on page 2-78.

2.2.2 ISPF for Sort

The RSVP Report Format panel displays when you enter F on the OPTIONS line on the RSVP Primary Selection Menu, or on the COMMAND line on any RSVP panel.

This panel is used for specifying RSVP keywords that control the fields to be sorted on and the report format. See Keywords for Sort on page 2-18 for a description of these keywords. The keywords for the report format are discussed in the next section.

RSVP Report Format

```
--------------------------  RSVP REPORT FORMAT  --------------------------
COMMAND ===>

LIST     ===>                       ALLOC UNITS     ===> TRKS

PRINT    ===> YES                     MERGE OUTPUT  ===> YES
OPTION   ===> NEW
FIELDS   ===> DSNAME ALLOC USED PCT HIALLRBA HIUSERBA NUMRECS
         ===>

SORT     ===> YES
FIELDS   ===> VOLUME
         ===>
DSNAME LENGTH ===> 44              SMS NAME LENGTH  ===>
INDEX TOTALS  ===>                 IPC COMMENT LEN  ===>
TOTALS        ===> YES             CHANGE CHARS     ===>
SUB TOTALS    ===>                 CHANGE CHARS     ===>

CHARS/LINE   ===> 133  LINES/PAGE ===> 60   BREAK ===>
HEADING  ===> YES
HEADING1 ===>  Find all DSNs containing ASM2DEMO
HEADING2 ===>  non-VSAM
```

2.3 Output Format Phase

The Output Format phase generates a report or output transaction file ($RSTRANS). This section covers report format only; transaction files are discussed in detail in Chapter 4 in the section Transaction Files on page 3-3.

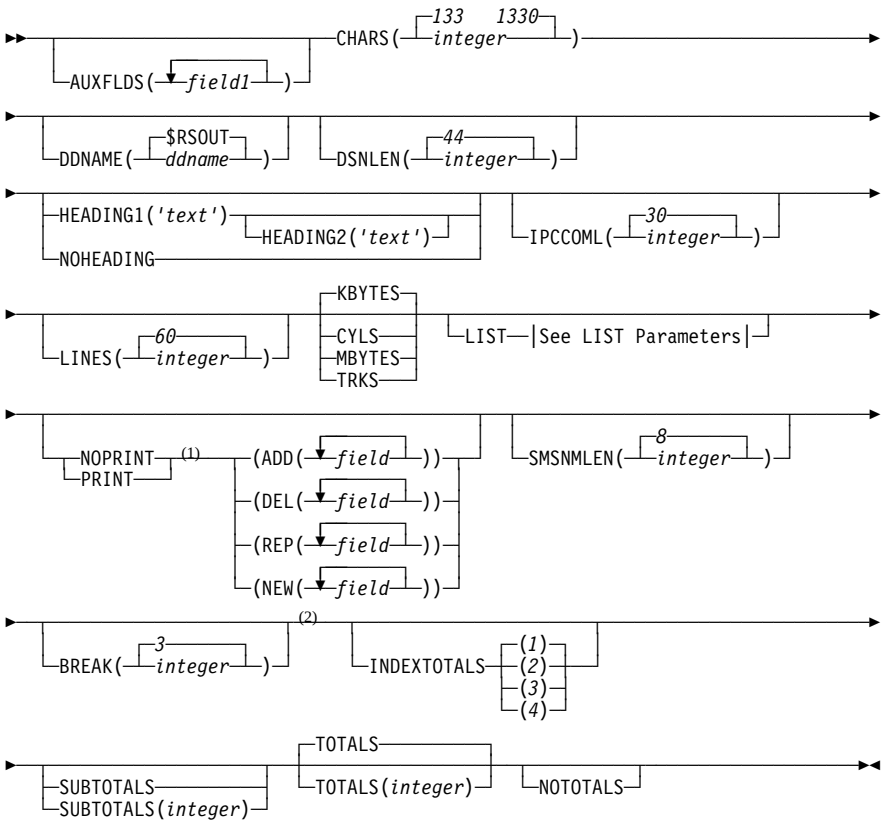
You can generate standard, modified-standard, and customized reports to provide exception, detail, or summary reporting on virtually any data set information you desire, and in the specific format you choose. The Output Format phase lets you specify the report layout and the fields to be printed. The LIST keyword option provides standard reports with fields that have been predefined in the user CSECT. You can use any of the defined lists. To produce modified-standard reports or customized reports with the fields of your choice, you can easily modify the list or define a new list.

Both the LIST and PRINT keywords control the fields to be printed. While LIST selects a predefined list of fields, the PRINT keyword modifies that list by adding, replacing, or deleting fields, or it creates a new list. The default list is the SPACE report.

The default heading line of RSVP output contains the string $RS0A03 RSVP VERSION followed by the version string, as much of the command line as can fit on the default heading line, and then the page number in the last 11 columns. If a selected field is not applicable with the selected input source, the field is left blank on the output report.

2.3.1 Command Format for Output Format

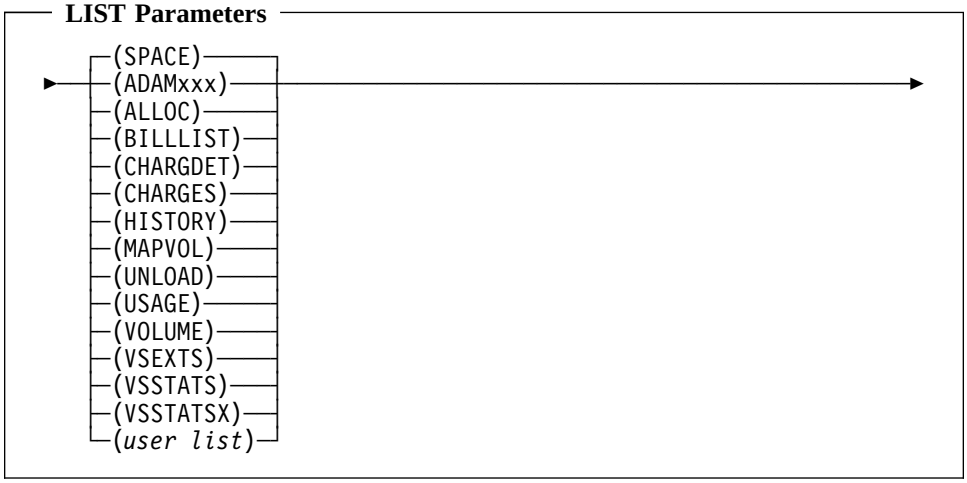
Command Format for Output Format



Notes:

1 Fields valid in the PRINT keyword are valid in NOPRINT to allow control of the subtotal and total print lines.

2 Defaults to length of SORT field.



Keyword Descriptions for Output Format

AUXFLDS Defines a list of auxiliary fields to be processed by the $RSVP command in addition to all the fields specified or defaulted for LIST (report field list), PRINT (print list), BLIST (billing field list), and IF (attribute selection).

For space and overhead reasons, only those primary fields needed to process the command itself are captured and formatted by default. User exits may need additional fields for other purposes, and those auxiliary fields can by specified by the AUXFLDS keyword.

CHARS Defines the logical record length (number of characters per line) and the block size. The default logical record length is 133 and the default block size is 1330. Both defaults can be changed in the user CSECT (see 5-9). The report output file $RSOUT is record format FBA.

You can specify either one value or two values separated by blanks for CHARS. If one value is specified, it is used for both record length and block size. If two are specified, the first is record length and the second is block size.

If you are consistently using a line size other than 133, change the user CSECT (see 5-9) to improve blocking. For example: CHARS(121) generates a line size of 120 (1 control character and 120 print positions), and CHARS(121 12100) generates a line size of 120 and a block size of 12100.

DDNAME Specifies the report DDNAME. The default is $RSOUT. The default can be changed in the user CSECT (see 5-9). All references in this text assume the default of $RSOUT.

DSNLEN Specifies the size of the DSNAME field on the report and output transaction file ($RSTRANS). The default is 44. Another value can be established in the user CSECT (see 5-9). For example: DSNLEN(20) truncates the data set name to 20 positions.

HEADING1 Replaces the normal heading on line 1 on a report. The heading must be in single quotes. The first position must contain an ASA print control character. A 1 in the first position indicates the HEADING1 line is to be on the top of a new page. Spacing is exactly as entered. For example: HEADING1('1MY REPORT') causes a page break with the heading and the report is titled MY REPORT. The $RSOUT DD statement must be present to obtain this heading. An example of a HEADING1 is shown in Using Descending Sort on page 2-79.

HEADING2 Defines a second title line. The format is identical to HEADING1. If HEADING2 is used, the current date is printed on the second title line below the page number. For example: HEADING2('-BY JOE BLOE') causes the second line to be spaced down three lines from the first and prints BY JOE BLOE followed by the current date. The $RSOUT DD statement must be present to obtain this heading. An example of a HEADING2 is shown in Using Descending Sort on page 2-79.

NOHEADING Eliminates all page breaks and heading lines, including the default.

IPCCOML Indicates the length of the comment field that is reported if the IPCCOM field is part of a print or billing list. The default is 30 characters.

LINES Defines the number of lines per page. The default is 60. The default can be changed in the user CSECT (see 5-9). For example: LINES(80) sets the number of lines per page to 80.

When reporting or billing on DASD space, the allocation and usage of space are often expressed in conflicting terms. For example, a data set on a 3390 taking up 10 tracks does not hold as much data as a 10-track data set on a 3390. The maximum for the data set on the 3390 is 474760 while the maximum for the 3390 is 566640. However, at other times you may want the allocation in terms of tracks and so forth. Any one of four possible representations for allocation are allowed:

CYLS Allocation is in cylinders. This has the same problems as TRKS and MBYTES (described next) because it is not meaningful on mixed devices, and rounding limits its usefulness for small data sets.

KBYTES Allocation is in kilobytes (thousands of bytes). KBYTES shows reasonable accuracy; it allows data sets on different device types to be represented in the same terms. KBYTES is the default. The default can be changed in the user CSECT (see 5-9).

MBYTES Allocation is in megabytes (millions of bytes). Rounding makes MBYTES less accurate on all but very large data sets.

TRKS Allocation is in tracks. Because of the differences in track sizes on different devices, TRKS is not an accurate unit of measurement when combining mixed device types.

LIST Specifies the report field list to be used. For example, LIST(USAGE). These report field lists were predefined in the user CSECT. The default report field list is SPACE. Observe that ADAM101 through ADAM610 reports (CA-3 reports) have been built into RSVP to provide even more report options. To use these reports, the appropriate keywords (VOL, IF, LIKE, and so on) must also be specified.

Valid list names and the fields to be printed are:

List	Fields to Print
ADAM101	VOLUME, DSNAME, DSORG, EXT, CCHH, SECT, SECQ, ALLOC, USED, PCT, DSCOUNT ADAM101 - Data Set Allocation Status Report provides detailed information on all the data sets that are allocated on the volumes being scanned. This report shows the type of data set, where it is located, and how much of the allocated space is being used.
ADAM102	VOLUME, DSNAME, STRACK, ETRACK, TRKALLOC, CCHH, FCYLS, EXTSEQ, EXT, DSCOUNT ADAM102 - Detail Track Allocation on Volume Report gives the physical layout of a volume. This information lets you determine whether the positioning of data sets on the volume is efficient, and whether space allocation on a

volume should be reassessed and reallocated for better system performance.

ADAM102V VOLUME, VOLSPC, VOLALLOC, VOLIXSP, VOLEXT

ADAM102V supplements the ADAM102 report by providing additional summary information about the volume.

ADAM103 VOLUME, DSNAME, EXT, DSORG, RECFM, BLKS2, LRECL, CREDT, EXPDT, LSTUS, PW, CAT, DSCOUNT

ADAM103 - Nontemporary Data Set Status Report lists the general physical layout and status of all nontemporary data sets on direct-access devices. This report lets you determine which data sets can be scratched, which data sets should be copied to tape, and which data sets occupy too many extents.

ADAM104 VOLUME, DSNAME, EXT, DSORG, RECFM, BLKS2, LRECT, CREDT, EXPDT, LSTUS, DSCOUNT

ADAM104 - Temporary Data Set Report lists the general physical layout and status of all temporary data sets on direct-access devices. This report lets you access temporary data set status. You can then initiate reorganization and scratching to maximize utilization and available space.

ADAM105 VOLUME, DSNAME, CREDT, EXPDT, LSTUS, LMDATE LMJOB, TRKALLOC, DSCOUNT

ADAM105 - Data Set Aging Status Report identifies temporary and nontemporary data sets that have not been accessed in a specified period of time. Using this information, you can determine which data sets are obsolete, not used, or unnecessary.

ADAM201 VOLUME, FRCYLS, FRTRKS, LFCYLS, LFTRKS, VOLEXT

ADAM201 - Available Free Space Report details the direct-access space available. Using this information, you can determine which volumes and device types have available space and how the space is divided on the volumes.

ADAM301 VOLUME, DSNAME, CREDT, EXPDT, LSTUS, LMDATE, ALLOC, DSCOUNT, CB

ADAM301 - Data Sets Selected for Controlled Scratch Report lists the direct-access data sets selected for scratch. Reviewing this report could help prevent valuable data sets (inadvertently flagged for scratch) from being scratched.

ADAM610 VOLUME, DSNAME, UNIT, UNLVOL, UNLFSEQ, CDATE, EDATE, IPCDATE, UNLRETPD, C3DSORG, BLKSIZE, ALLOC, USED, IPCUTYPE

ADAM610 - Migration/Backup Forecast Report lists the direct-access data sets selected for migration/backup. Reviewing this report could help prevent data sets (inadvertently flagged for migration/backup) from being migrated or backed up, or, conversely, identify data sets not selected to be migrated or backed up.

The next five reports, ALLOC, BILLLIST, CHARGDET, CHARGES, and HISTORY are designed for the RSVP billing facility. For an example showing the use of the CHARGDET and CHARGES report lists and their output, see Two-Step Billing on page 4-11.

ALLOC DSCOUNT, ALLOC, USED, UNUSED, PCT, EXT, SECQ, SECT, DSNAME, VOLUME, DSORG, RECFM, BLKSZ, LRECL, CREDT, LSTUS

BILLLIST DATE, UNIT, VOLUME, DSNAME, ACCOUNT, KBALLOC, KBDAYS, CHCUR

CHARGDET ACCOUNT, DSCOUNT, KBALLOC, KBDAYS, CHCUR, CHMTD, CHYTD, UNIT, VOLUME, DSNAME

The CHARGDET report shows billing details. For an example, see Two-Step Billing on page 4-11.

CHARGES ACCOUNT, DSCOUNT, ALLOC, KBDAYS, CHCUR, CHMTD, CHYTD

HISTORY DATE, CHYTD, CHMTD, CHMONTHS, ACCOUNT

MAPVOL DSCOUNT, DSNAME, ABSTR, ABLEN, ALLOC, CCHH, EXTSEQ, EXT, VOLUME

The MAPVOL report provides information on

the general physical layout of specific data sets on the volume.

SPACE DSCOUNT, DSORG, LSTUS, ALLOC, USED, UNUSED, EXT, VOLUME, DSNAME

The SPACE report is the default LIST report that shows the space allocations for data sets. For an example of the SPACE report, see Example 1 on page 2-39.

UNLOAD IPCDATE, IPCUTYPE, IPCHVOL, IXRCAND, UNLVOL, UNLFSEQ, DSNAME

The UNLOAD report is an IPC report that provides specific backup and reload information on data sets.

USAGE DSCOUNT, USECNT, LSTUS, LMDATE, LMTIM, LMJOB, VOLUME, ALLOC, DSNAME

The USAGE report is a summary of data set usage information.

VOLUME VOLUME, CUU, UNIT, DEVTYPE, VOLATTR, VOLSPC, VOLEXT, VOL1XSP, DSREC, DSTOT, VTOCI, CYLPVOL, TRKPCYL, DEVTK, DEVDT, DEVDB, VTOCE, VTOCL, VOLLSPC

The VOLUME report is a summary of volume information. For an example VOLUME report, see Online Volume Free Space on page 2-85.

The next three VSAM reports, VSEXTS, VSSTATS, and VSSTATSX, show extent and statistic information on VSAM data sets.

VSEXTS DSNAME, HIALLRBA, HIUSERBA, #EXLOCEH, EXLOCCH(1:2), EXHICCH(1:2)

For multivalued fields EXLOCCH and EXHICCH, the 1:2 indicates a range of values. In this case, the first and second values of each field are reported.

VSSTATS DSNAME, CIPCA, PCA, PCI, NUMRECS, RETRECS

VSSTATSX DSNAME, CIPCA, PCA, PCI, NUMRECS, DELRECS, INSRECS, UPDRECS, RETRECS, CISPLITS, CASPLITS, NUMEXCPS

Note: If the print line cannot hold all the fields defined in a list (the print-line length exceeds the CHARS value), ending fields with lengths exceeding 18 are

truncated and shorter fields are ignored. Since the print lists are defined in the user CSECT, it is easy to add new print lists or modify the current ones. See User CSECT on page 5-9 for information on modifying and creating print lists.

PRINT Modifies or replaces the print list. For multivalued field names, you can express a range of values to be part of the list by using a colon to separate the field values. For example, HIALLRBA(1:5) prints the first through fifth values of HIALLRBA.

Multivalued fields may contain a variable number of entries in the same record. To list each of these values, a subparameter must be appended to the desired keyword. The subparameter is a low and high range number separated by a colon and indicates the range of entries to be printed. For example, the EXTRKS field displays the number of tracks in an extent of a VSAM file. To list each of the EXTRKS of a VSAM file with ten extents, use EXTRKS(1:10).

Over 500 fields are in the active print list for RSVP. These fields are listed in alphabetical order in Appendix A.

There are four options for using the PRINT keyword:

ADD Adds fields to the print list. Following ADD, enter a list of fields. The first field entered must be a field that is in the active print list. Insert the additional fields listed after the first field. For example: PRINT(ADD (LSTUS LMDATE LMTIM LMJOB)) inserts last modified date (LMDATE), time (LMTIM), and job (LMJOB) into the print list following last use date (LSTUS). See Example 1.2 Using ADD on page 2-81.

DEL Deletes fields in the current print list. If only one field is being deleted, it is not necessary to put it into parentheses, but multiple fields must be in parentheses. For example: PRINT(DEL DSCOUNT) deletes the data set count, while PRINT(DEL (DSCOUNT UNUSED LSTUS)) deletes the data set count (DSCOUNT), the unused space (UNUSED), and the last used date (LSTUS). See Example 1.1 Using DEL on page 2-80.

REP Replaces a field in the print list with other fields. The format is the same as ADD, but the first field listed is deleted from the list. For example: PRINT(REP (LSTUS LMDATE LMTIM LMJOB)) replaces last use date (LSTUS) with last modified date (LMDATE), time (LMTIM), and job (LMJOB). See Example 1.3 Using REP on page 2-82.

NEW Ignores the print list and creates a new print list using the fields in parentheses. If there is only one field, it is not necessary to put it into parentheses, but multiple fields must be in parentheses. For example: PRINT(NEW

(ALLOC PCT DSORG DSNAME)) generates a report with the fields allocation (ALLOC), percent used (PCT), data set organization (DSORG), and data set name (DSNAME). See Example 1.4 Using NEW on page 2-83.

Note: When using the PRINT keyword, a blank must follow the ADD, REP, DEL, and NEW operands.

NOPRINT Eliminates the printing of detail lines. However, the fields valid in the PRINT keyword (ADD, REP, DEL, and NEW) are valid in NOPRINT to allow control of the subtotal and total print lines. For an example using NOPRINT, see Example 2.1 on page 2-84.

Note: If NOPRINT is specified, you must specify SORT if using INDEXTOTALS or SUBTOTALS. NOPRINT causes a default of NOSORT.

SMSNMLEN Indicates the length of the SMS construct name:

`DATACLASS, MGTCLASS, STGROUP, STRCLASS.`

The default is 8 characters. The maximum length is 30 characters. An example using SMSNMLEN is shown on 3-19.

Additional keywords allow you to make the report easier to read or more meaningful for your installation. You can define subtotaling, index totaling, and page breaks.

BREAK Forces page breaks when the primary sort field changes. An integer in parentheses following BREAK indicates the number of meaningful characters in the primary sort field to compare for a change. Whenever these characters change, RSVP forces a page break. The default is the length of the primary sort field. For example: SORT(VOLUME DSNAME) BREAK causes a page break when the volume changes (after the subtotal is printed if SUBTOTALS was also specified), while SORT(DSNAME) BREAK(4) causes a page break when the first four characters of the DSNAME changes. If INDEXTOTALS is also specified, the break is on the high-level index. The $RSOUT DD statement must be present to obtain the break.

INDEXTOTALS Provides subtotals on the high-level qualifiers of the DSNAME. It is valid only if DSNAME is the first sort field; if not, SUBTOTALS is assumed instead of INDEXTOTALS. An integer in parentheses following INDEXTOTALS can be a value of 1, 2, 3, or 4 to indicate the index level on which it subtotals. The default is 1, indicating a subtotal on the first index level. For example, SORT(DSNAME) INDEXTOTALS gives subtotals each time the high-level qualifier changes, while SORT(DSNAME) INDEXTOTALS(2) gives subtotals whenever the second-level qualifier changes and when the high-level qualifier changes. If the SORT keyword is not specified, SORT(DSNAME) is assumed (the

default). For an example using INDEXTOTALS, see Example 2.1 on page 2-84.

SUBTOTALS Provides subtotaling of numeric data set and VSAM fields whenever the primary sort field changes. An integer in parentheses following SUBTOTALS indicates the number of meaningful characters in the primary sort field to compare for a change. Whenever these characters change, it is subtotaled. When BREAK is not specified, the default is the length of the primary sort field. When BREAK is specified, the default gives a subtotal at each forced page break. For example: SORT(VOLUME DSNAME) SUBTOTALS gives a subtotal each time the volume changes, while SORT(DSNAME) SUBTOTALS(4) gives a subtotal every time any of the first four characters of DSNAME changes.

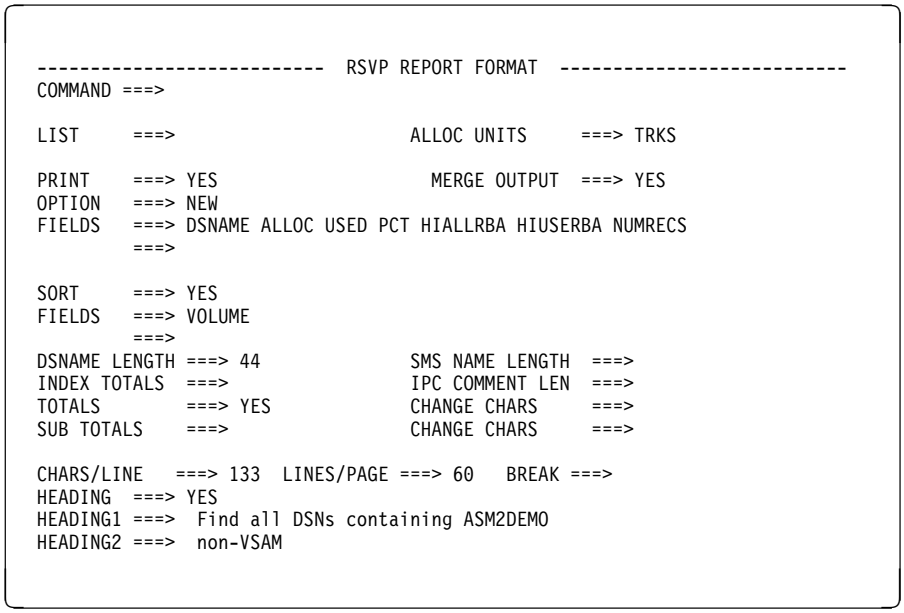
TOTALS Provides totaling of numeric data set and VSAM fields whenever the primary sort field changes. An integer in parentheses following TOTALS indicates the number of meaningful characters in the primary sort field to compare for a change. Whenever these characters change, a total is given. For example: SORT(VOLUME DSNAME) TOTALS gives a total each time the volume changes, while SORT(DSNAME) TOTALS(4) gives a total every time any of the first four characters of DSNAME changes. TOTALS is the default.

NOTOTALS Eliminates all subtotals and the final totals.

2.3.1.1 ISPF for Output Format

The RSVP Report Format panel displays when you enter F on the OPTIONS line on the RSVP Primary Selection Menu, and on the COMMAND line on any RSVP panel.

This panel lets you specify keywords that control the report format by identifying report layout, fields to be sorted on, fields to be printed, and so on. For a description of these keywords, see Keyword Descriptions for Output Format on page 2-22.

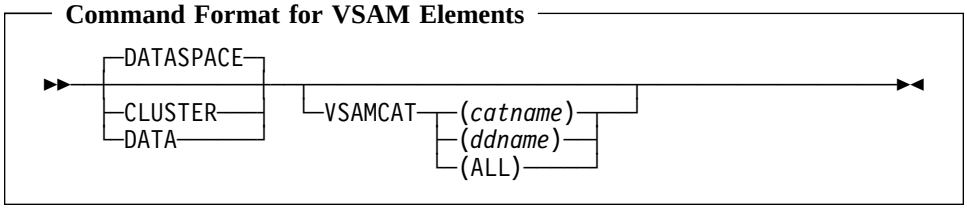


2.4 VSAM Reporting

RSVP can report on VSAM elements on a volume with standard non-VSAM data sets. It reports VSAM elements in one of three forms: data space level, cluster level, or component level. The default DATASPACE reports on information contained in the DSCB in the VTOC.

You can list Master Catalogs, user catalogs, page spaces, swap spaces, alternate indexes and clusters. You can change the user CSECT (see 5-9) to control the various elements actually reported on, and, in addition, test the VS (VSAM entity type) field in Boolean selection to eliminate the unwanted element types.

2.4.1 Command Format for VSAM Elements



Keyword Descriptions for VSAM Elements:

For examples showing the use of these keywords, see VSAM Reporting Examples on page 2-88.

CLUSTER Specifies that VSAM entities are to be selected and reported at the cluster level rather than at the component level. When CLUSTER is specified, the field VS contains:

C Cluster entity
G Alternate index
M Master catalog
U User catalog
P Page space
S Swap space

The $RSVP command must include VOLUME and CLUSTER for the USED field on the report to represent the space containing VSAM data. If VOLUME is not included, the USED field represents allocation tracks for VSAM data sets and the UNUSED field is always zero.

DATA Specifies VSAM entities are to be selected and reported at the component level rather than at the cluster level. When DATA is specified, the field VS contains either a D for data elements or an I for index elements.

The $RSVP command must include VOLUME and DATA for the USED field on the report to represent the space containing VSAM

data. If VOLUME is not included, the USED field represents ALLOC tracks for VSAM data sets and UNUSED is always 0.

DATASPACE Specifies that VSAM data sets are to be reported only by the information contained in the DSCB in the VTOC. DATASPACE is useful for identifying VSAM components quickly. DATASPACE is the default. If desired, you can change the default in the user CSECT (see 5-9).

The $RSVP command must include VOLUME and DATASPACE for the USED field on the report to represent the space containing VSAM data. If VOLUME is not included, the USED field represents allocation tracks for VSAM data sets and the UNUSED field is always 0.

VSAMCAT Specifies which BCSs or VSAM catalogs are to be searched for clusters and alternate indexes found on the volumes being searched. If VSAMCAT is not specified, RSVP searches the default catalog associated with the JOB step. For VSAMCAT, the values are:

1. A VSAM catalog name: VSAMCAT(USER.CAT.NAME)
2. The name of a DD that defines the VSAM catalog to be searched: VSAMCAT(CATDD)
3. The keyword ALL which means to search all VSAM catalogs that can be found in the system where the CA-ASM2 command processor is running: VSAMCAT(ALL)

However, avoid using VSAMCAT(ALL) if possible. It creates extremely poor performance and can result in response times of minutes rather than seconds, depending on your environment. See Performance Notes on page 5-33 for more information.

For example,

```
VSAMCAT(CATALOG.USER1) CLUSTER
```

reports on clusters on the volumes specified as long as they are in CATALOG.USER1. The statement

```
//CATDD DD DSN=CATALOG.USER,DISP=SHR
```

with VSAMCAT(CATDD) DATA lists the data and index elements of VSAM data sets on the specified volumes as long as they are in CATALOG.USER1.

When listing VSAM, the field BLKSZ has CISIZE and the field LRECL has maximum record length.

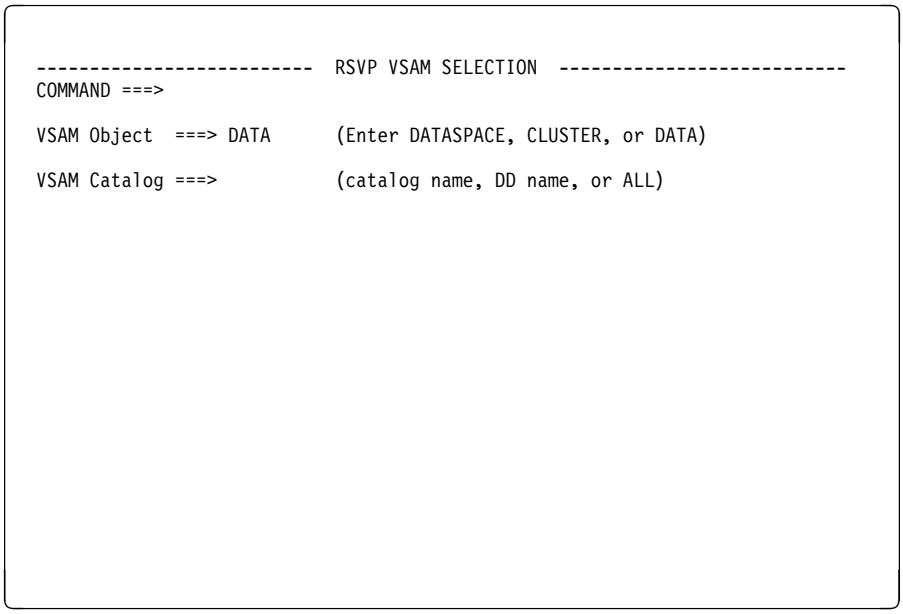
For examples using VSAMCAT, see Using VSAMCAT to Reference a DDNAME on page 2-88 and Using VSAMCAT to Reference a VSAM Catalog on page 2-89.

2.4.1.1 ISPF for VSAM Selection

The RSVP VSAM Selection panel displays when you enter V on the OPTION line on the RSVP Primary Selection Menu, or on the COMMAND line on any RSVP panel.

This panel lets you control the search for VSAM elements. See Keyword Descriptions for VSAM Elements on page 2-32 for a description of the keywords.

RSVP VSAM Selection



2.5 User Exit Descriptions

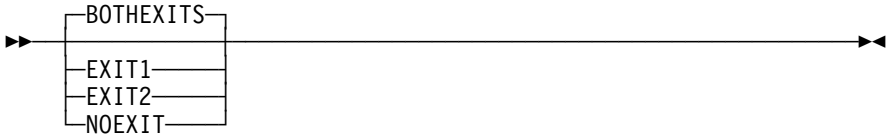
Two standard user exits are provided, a Selection exit and a Post-Sort exit for controlling the $RSVP command process. The default names for these exits are $RSEXT1 and $RSEXT2, respectively.

$RSEXT1, the Selection exit, lets you include or exclude data sets based on an optional input file referenced by the $RSEXT1I DD. This exit receives control before a data set is finally selected, and has the option of rejecting the data set. $RSEXT2, the Post-Sort exit, receives control after the sort. This exit lets you modify the output after the sort. Figure 3 on 2-36 illustrates the control points for both exits.

Both $RSEXT1 and $RSEXT2 can be modified to your installation's requirements. See RSVP User Exits on page 5-4 for more information. The default load names of these exits can be changed in the user CSECT. With multiple user CSECTs, it is possible to invoke multiple exits. See User CSECT on page 5-9 for more information.

2.5.1 Command Format for User Exits

Command Format for User Exits



Keyword Descriptions for User Exits

EXIT1 Causes the Selection exit $RSEXT1 to execute.

EXIT2 Causes the Post-Sort exit $RSEXT2 to execute.

BOTHEXITS Causes both $RSEXT1 and $RSEXT2 exits to execute. BOTHEXITS is the default.

NOEXITS Eliminates invocation of all exits.

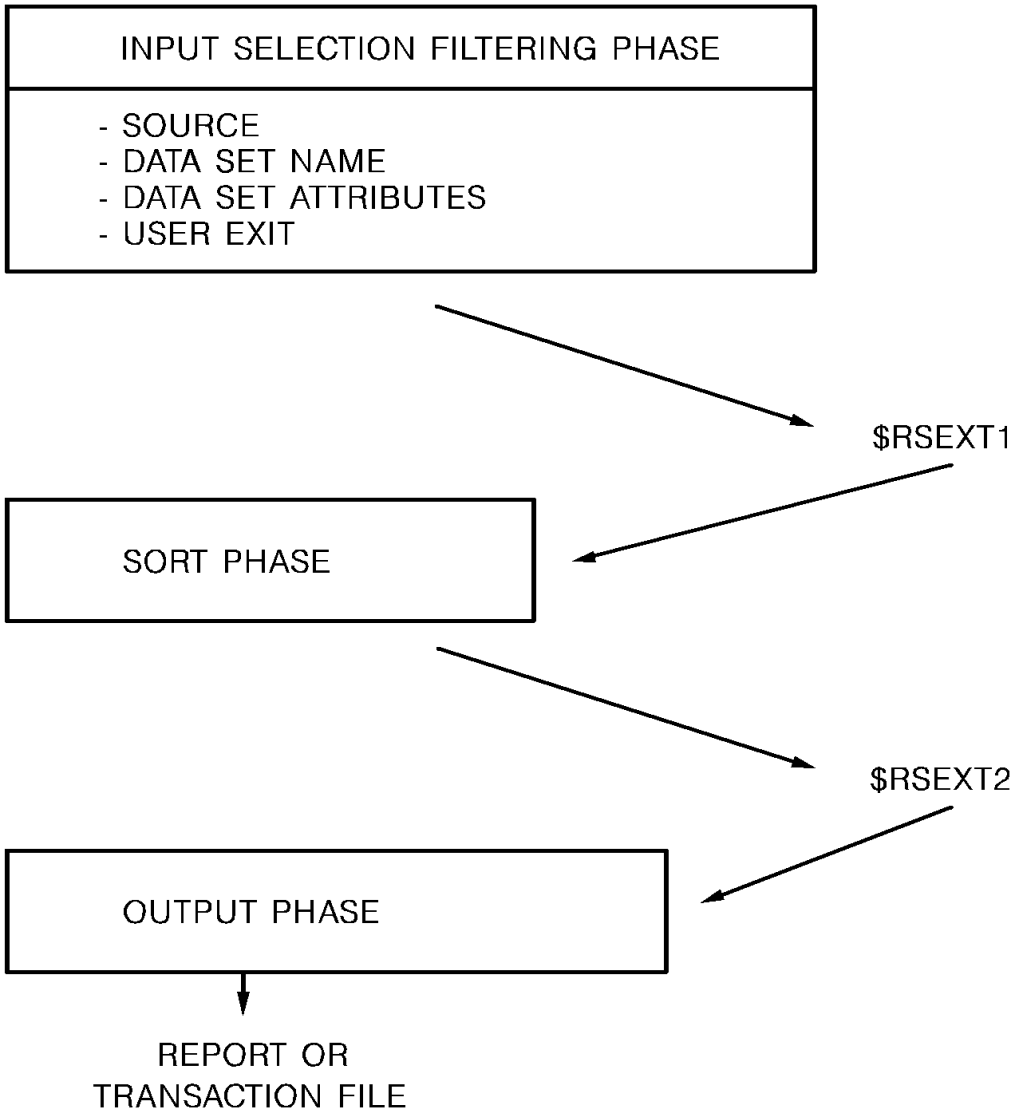


Figure 3 - RSVP User Exits

2.5.2 Sample Input File for EXIT1

The input file for EXIT1, Selection exit $RSEXT1 is referenced by DDNAME $RSEXIT1I. An example of the input stream follows. You can insert comments by putting an asterisk (*) in column 1. For example, *ROD**.-.DAILY is a comment.

```
//$RSEXT1I DD *
&E                  Exclude the following:
SYS1.              All system data sets (beginning with SYS1)
SYSCTLG             CVOL catalogs
-.LIST              TSO list data sets
-.OUTLIST           TSO output list data sets
PROD***.-.DAILY     Daily production data sets
*                   (according to shop standards)
PROD***.-.TEMP.-    Temporary production data sets
/*
```

where,

$RSEXT1I Is input to EXIT1, the Selection Exit.

2.5.2.1 Control Statements

The following control statements select or exclude specific data sets using $RSEXT1. In each case, the & must begin in column 1. When entering data set names, levels, or pattern masks, also begin in column 1.

Parameter Description

&I Include only data sets that meet the selection criteria of this exit and the selection criteria of the command. A flag exists in the user CSECT (see 5-9) that specifies the exit can override the command on selection. Additional control statements with data set names, pattern masks, or beginning character strings follow the statement.

&E Exclude data sets from selection even if the command would select them. Statements following this are the same as those following &I.

&O Reverse the decision of the command for the data sets, levels, or pattern masks following. If the command selects a data set, exclude it. If the command rejected a data set, select it. EXITFLG=ALWAYS should be specified in the user CSECT (see 5-9) to allow this exit to make the final decision; using &O rejects everything. Statements following this are the same as those following &I.

Note: &I, &E, and &O are mutually exclusive. All data set names specified are processed using the last action specified.

2.5.3 ISPF for User Exits

The RSVP User Exits panel displays when you enter U on the OPTIONS line on the RSVP Primary Selection Menu, or on the COMMAND line on any RSVP panel.

This panel lets you control the user exit routines for tailoring the $RSVP command. See Keyword Descriptions for User Exits on page 2-35 for a description of the keywords, and User CSECT on page 5-9 for a description of the user CSECT.

RSVP User Exits

```
------------------------------  RSVP USER EXITS  ------------------------------
 COMMAND ===>

 Invoke Selection Exit ===> NO

 Invoke Post-Sort Exit ===> NO

 Customized User CSECT ===> $RSUSER  (load module name)
```

2.6 Examples

This section provides several selection examples for the following:

- DASD volume
- SMS
- OSCAT
- ASM2CAT
- TMC
- VMF
- Combined
- Sort
- Output format
- VSAM reporting

2.6.1 DASD Volume Selection Examples

2.6.1.1 Example 1 - Listing a User's Data Sets

The simplest use of the $RSVP command is to list a user's data sets. In this example, since the LIST keyword is not specified, the default SPACE report field list is used.

```
$RSVP VOL(*) LEVEL($CAI)
```

This command generates the following standard report:

```
$RS0A03 RSVP VERSION 1.0     $RSVP                                          PAGE 1
DSCOUNT DSORG  LSTUS     ALLOC   USED   UNUSED  EXT  VOLUME   DSNAME

         PO    06/11/00  1335    1144   191     8     USER01  $CAI.M21.CDS64
         PO               400      19   381     1     TEST01  $CAI.M21.CDS65
         PO    05/31/00    95      38    57     1     USER01  $CAI.SYSQA.CACI
         PO    06/09/00   248     229    19     6     USER01  $CAI.SYSQA.CACA
         PO    06/10/00   400     210   190     3     USER01  $CAI.V67.LOOK.OBJECT
         PS    05/30/00    19      19     0     1     USER01  $CAI.V67.REF

         |       |        |       |     |      |       |          |

         PO    05/31/00   191      95    96     1     USER01  $CAI.DOC.SRG
         PS    06/09/00    57      57     0     1     USER02  $CAI.DOC.GIG
         PO    06/09/00    19      19     0     1     USER01  $CAI.TEST.MACROS
         PS    06/10/00    57      38    19     1     TEST01  $CAI.TEST.SYS1
         PS    06/01/00    38      38     0     1     USER02  $CAI.A24.ROSCOE
         PO    06/10/00    38      19    19     1     USER01  $CAI.A25.ROSCOE
    23                    19943   12964  6959    60           TOTAL
```

Field Descriptions

DSCOUNT	Data set count.
DSORG	Data set organization. PO=Partitioned and PS=Physical Sequential.
LSTUS	Last date the data set was referenced.
ALLOC	Total allocation quantity for the data set.
USED	Quantity of space used. KBYTES is the default.
UNUSED	Quantity of unused space.
EXT	Number of extents in the data set.
VOLUME	Volume serial number.
DSNAME	Name of the data set.

2.6.1.2 Example 2 - Using IF/AND/ANDIF/ORIF Keywords

This example selects data sets according to specific attributes:

```
$RSVP VOL(USER) PRINT(DEL (DSCOUNT VOLUME)) -
IF(DSORG EQ PO) -
AND1(ALLOC GT 100) -
AND2(EXT GT 1) -
ORIF3(DSORG EQ PS) -
AND4(ALLOC GT 20) -
ANDIF5(LSTUS LT *-5) -
TRK
```

This allows for two possible combinations of attributes:

1. A PDS with more than 100 tracks allocated, having multiple extents, and not used in the last five days (keywords IF, AND1, AND2, and ANDIF5); or
2. A sequential data set with more than 20 tracks allocated, and not used in the last five days (keywords ORIF3, AND4, and ANDIF5).

The keyword TRK is specified so the field ALLOC is reported in tracks rather than KB.

This command generates the following customized report:

```
$RS0A03 RSVP VERSION 1.0     $RSVP VOL(USER)       PAGE  1
DSORG LSTUS    ALLOC USED UNUSED  EXT DSNAME

 PO   06/01/00   150   84     66    5 $CAI00.STEPLIB
 PS   05/19/00    30    1     29    1 $CAI05.R25.DATA
 PS   06/02/00   300  300      0    1 $CAI05.SOURCE.PAN
 PS   05/19/00    48   48      0    1 $CAI99.$ARVSAM.LIST
 PO   05/19/00   126  116     10    2 $CAI99.IBMMAC.ASM
 PO   05/21/00   124  116      8    2 $CAIAI.$RS.IBMMAC
 PO   06/03/00   120   99     21    3 $CAI99.ARCH.MANUAL
                 898  764    134   15        TOTAL
```

Field Descriptions

DSORG Data set organization. PO=Partitioned and PS=Physical Sequential.

LSTUS Last date the data set was referenced.

ALLOC Allocation quantity for the data set.

USED Quantity of space used. KBYTES is the default.

UNUSED Quantity of unused space.

EXT Number of extents in the data set.

DSNAME Name of the data set.

2.6.1.3 Example 3 - Using Data Set Name Selection Keywords

Example 3.1 Using LEVEL, NOTLAST, and CONTAIN

This command selects all data sets with high-level indexes (LEVEL) of $CAI00 or $CAI03 that do not have last-level indexes (NOTLAST) of LIST, OUTLIST, or LINKLIST as long as they contain (CONTAINING) the characters ARCH.

```
$RSVP VOL(USER) TRK PRINT(DEL (DSCOUNT DSORG VOLUME)) -
LEVEL($CAI00 $CAI03) -
NOTLAST(LIST OUTLIST LINKLIST) -
CONTAIN(ARCH)
```

This generates the following customized report:

```
$RS0A03 RSVP VERSION 1.0     $RSVP VOL(USER)      PAGE  1
LSTUS    ALLOC USED UNUSED EXT DSNAME

06/01/00     5    3      2   1 $CAI00.ARCH.CAI.CNTL
05/19/00    28    8     20   1 $CAI00.ARCH.INSTALR.CNTL
05/19/00     4    4      0   1 $CAI00.ARCH.RACF.MAC.ASM
05/19/00    18    7     11   1 $CAI00.ARCH.TESTR.PROCLIB
06/03/00     2    1      1   1 $CAI03.ARCH.CNTL
06/01/00   150    6    144   1 $CAI03.ARCH.DATA
06/02/00    10    5      5   1 $CAI03.ARCH.INSTALLG
06/11/00   150  129     21   1 $CAI03.ITP.ARCH.DATA
05/21/00    54   54      0   1 $CAI03.ITP.ARCH.LOADTEMP
06/03/00   120   99     21   3 $CAI03.ITP.ARCH.MANUAL
           541  316    225  12        TOTAL
```

Field Descriptions

LSTUS Last use date.

ALLOC Data set allocation quantity.

USED Quantity of space used.

UNUSED Quantity of unused space.

EXT Number of extents in the data set.

DSNAME Data set name.

Example 3.2 Using BEGIN, NOTLEVEL, ENDING, and NOTLAST

This command selects any data set that begins (BEGIN) with the characters $CAI, that ends (ENDING) with the characters LIST except data sets with high-level index (NOTLEVEL) of $CAI02 or data sets with a low-level index (NOTLAST) of CLIST.

```
$RSVP VOL(USER) TRK PRINT(DEL DSCOUNT)) -
BEGIN($CAI) -
NOTLEVEL($CAI02) -
ENDING(LIST) -
NOTLAST(CLIST)
```

This generates the following customized report:

```
$RS0A03 RSVP VERSION 1.0     $RSVP VOL(USER)          PAGE  1
DSORG LSTUS  ALLOC USED UNUSED EXT VOLUME DSNAME

 PS   06/09/00  57   40     17   3 USER02 $CAI00.$AI.LIST
 PS   06/09/00  25   24      1   2 USER02 $CAI00.$CR.LIST
 PS   06/09/00  32   32      0   2 USER02 $CAI00.$DA.LIST
 PS   06/09/00   9    9      0   1 USER02 $CAI00.$DR.LIST
 PS   06/09/00  21   21      0   2 USER02 $CAI00.$PS.LIST
 PS   05/19/00   1    1      0   1 USER02 $CAI00.TESTLIST
 PS   05/27/00  16    6     10  16 USER02 $CAI00.ZAPLIST
 PS   05/19/00   1    1      0   1 USER02 $CAI00.T.TESTLIST
 PS   06/08/00  40   40      0   3 USER02 $CAI00.NAI.LIST
 PS   05/19/00  11    1     10   2 USER02 $CAI00.V.TESTLIST
 PS   05/27/00  11    1     10   2 USER02 $CAI00.S.TESTLIST
 PS   05/19/00  36   36      0   2 USER02 $CAI03.$MON.LIST
 PS   05/19/00   8    8      0   8 USER02 $CAI03.MISC.LIST
 PO   06/08/00 128  125      3   2 USER01 $CAI04.MIG.LIST
 PS   05/29/00  11    2      9   2 USER02 $CAI04.W.TESTLIST
               30   30      0   1 USER02 $CAI05.DUMP.LIST
 PS   05/19/00   4    4      0   1 USER02 $CAI99.GEN.LIST
 PO   05/19/00 100   25     75   1 USER02 $CAI99.VSAM.LIST
              563  414    149  55        TOTAL
```

Field Descriptions

DSORG Data set organization.

LSTUS Last use date.

ALLOC Data set allocation quantity.

USED Quantity of space used.

UNUSED Quantity of unused space.

EXT Number of extents in the data set.

VOLUME Volume serial number.

DSNAME Data set name.

2.6.1.4 Example 4 - Using Data Set Name Simple Pattern Masking

This command shows the use of simple pattern masking:

```
$RSVP VOL(SYS500) PRINT(DEL (DSCOUNT DSORG))  -
LIKE(-.$RS.-  -.CLIST  $CAI**.-.L79-.-)
```

This selects data sets that match **any** of three pattern masks defined by the LIKE keyword:

- -.$RS.- matches any data set with an index of $RS, preceded and followed by zero or more other indexes.
- -.CLIST matches any data set with a low-level index of CLIST preceded by any number of other levels.
- $CAI**.-.L79-.- matches any data set beginning with $CAI followed by up to two characters in the first index level, with a level beginning with L79 and any number of levels in between it and the first level, and any number of levels following it.

This generates the following customized report:

```
$RS0A03 RSVP VERSION 1.0   $RSVP VOL(SYS500)        PAGE  1
LSTUS   ALLOC  USED UNUSED EXT VOLUME DSNAME

06/11/00 3432  2975    457   2 SYS500 $CAI01.$RS.MACS.ASM
06/09/00 7437  6617    820   3 SYS500 $CAI01.$RS.PRINTAA
06/11/00  572    57    515   1 SYS500 $CAI05.$RS.DATA
06/11/00  572    38    534   1 SYS500 $CAI05.CMDPROC.CLIST
06/19/00   19    19      0   1 SYS500 $CAI99.$RS.CNTL
06/11/00  381   153    228   1 SYS500 $CAI99.CMDPROC.CLIST
06/19/00   19    19      0   1 SYS500 $CAI99.COMP.CLIST
06/21/00 1335  1049    286   1 SYS500 $CAI99.ITP.$RS.ASM
06/21/00 2365  2212    153   2 SYS500 $CAI99.ITP.IBMMAC
06/21/00  362   191    171   2 SYS500 $CAI99.ITP.$RS.LOAD
06/21/00   57    57      0   1 SYS500 $CAI99.ITP.MANUAL
06/10/00 6295  4786   1507   7 SYS500 $CAI99.L7910.ASM
06/10/00 3070  2822    248   3 SYS500 $CAI99.L7910.DATA
06/10/00 1144   629    515   2 SYS500 $CAI99.L7910.OBJ
06/19/00 2479  2288    191   1 SYS500 $CAI99.L7910.S.DATA
06/19/00 2288  1773    515   1 SYS500 $CAI99.L7910.S1.ASM
06/10/00 1526   648    878   6 SYS500 $CAI99.L7910MAC.ASM
06/19/00 1316  1316      0   1 SYS500 $CAI99.MAY3.$RS.ASM
06/19/00   19    19      0   1 SYS500 $CAI99.STATE.CLIST
        34686 27668   7018  38        TOTAL
```

Field Descriptions

LSTUS Last use date.

ALLOC Data set allocation quantity.

USED Quantity of space used.

UNUSED Quantity of unused space.

EXT Number of extents in the data set.

VOLUME Volume serial number.

DSNAME Data set name.

2.6.1.5 Example 5 - Listing VSAM Activity Statistics

This command lists activity statistics for VSAM data sets on volumes MVXE74 and SMS602.

```
$RSVP VOLUME(MVXE74 SMS602) CLUSTER IF(DSORG EQ VS) -
PRINT(NEW (DSORG VOL CISPLITS CASPLITS CIPCA -
INSRECS DELRECS UPDRECS NUMRECS DSN))
```

This command generates the following customized report:

```
VSAM ACTIVITY STATISTICS LIST                                                    PAGE 1
DSORG VOLUME   CISPLITS   CASPLITS  CIPCA    INSRECS    DELRECS    UPDRECS    NUMRECS DSNAME

 VS   MVXE74          5          0     10        407        433          1         45 ASM2.V400.TABLE
 VS   MVXE74          5          2     10        407        433          1         45 ASM2.V400.TABLE.DATA
 VS   MVXE74          0          0     46          0          0          5          1 ASM2.V400.TABLE.INDEX
 VS   SMS602          0          0     54         24          0        141        111 SMSDS.ASMAST.TABLE
 VS   SMS602          0          0     54         24          0        141        111 SMSDS.ASMAST.TABLE.D
 VS   SMS602          0          0     31          0          0         11          1 SMSDS.ASMAST.TABLE.I
 VS   SMS602          0          0      7          0          0          0         12 SMSDS.IXRC.T
 VS   SMS602          0          0      7          0          0          0         12 SMSDS.IXRC.T.D
 VS   SMS602          0          0     46          0          0          0          1 SMSDS.IXRC.T.I
 VS   MVXE74          3          1     20          4          2          1        114 VN9000.ASMAST.TABLE
 VS   MVXE74          3          1     20          4          2          1        114 VN9000.ASMAST.TABLE.DATA
 VS   MVXE74          0          0     46          0          0          0          1 VN9000.ASMAST.TABLE.INDEX
 VS   MVXE74          0          0    140          0          0          0       2923 VN9000.CSGADC2.SDDS
 VS   MVXE74          0          0    140          0          0          0       2923 VN9000.CSGADC2.SDDS.DATA
 VS   MVXE74          0          0     18          0          0          0         10 VN9000.CSGADC2.SDDS.INDEX
                     16          4    649        870        870        302       6424 TOTAL
```

Field Descriptions

DSORG Data set organization. VS is VSAM.

VOLUME Volume serial number.

CISPLITS Number of control interval splits for the data set.

CASPLITS Number of control area splits for the data set.

CIPCA Number of control intervals in a control area for the data set.

INSRECS Number of records inserted in the data set.

DELRECS Number of deleted records in the data set.

UPDRECS Number of updated records in the data set.

NUMRECS Number of user-supplied records in the data set.

DSNAME Data set name.

2.6.2 SMS Selection Examples

2.6.2.1 Example 1 - Listing All Volumes in SMS Storage Groups

This command searches SMS Storage Groups for all volumes. It lists the volumes with some of their attributes.

```
$RSVP STGROUP(*) PRINT(NEW (VOLUME STGROUP VOLALLOC STGSTAT -
STGTHRSH STGTYPE STGABKP STGABSYS STGADMP)) SMSNMLEN(10)
```

The command generates the following customized report:

```
$RS0A03 RSVP VERSION 1.0     $RSVP  STGROUP(*)                          PAGE 1
VOLUME  STGROUP     VOLALLOC  STGSTAT  STGTHRSH  STGTY  STGABKP  STGABSYS  STGADMP

SMS601  SGTEST          2849  ENABLED  THRESHOLD POOL   AUTOBKP  XE74      AUTODMP
SMS602  SGTEST          7121  ENABLED  THRESHOLD POOL   AUTOBKP  XE74      AUTODMP
```

Field Descriptions

VOLUME Volume serial number.

STGROUP Name of the Storage Group.

VOLALLOC Amount of allocated space on the volume.

STGSTAT Status of the Storage Group: NONE, ENABLED, QUI/ALL, QUI/NEW, DIS/ALL, DIS/NEW.

STGTHRSH Threshold status of the Storage Group.

STGTY (STGTYPE) Type of Storage Group: POOL, VIO, DUMMY.

STGABKP Automatic backup status of the Storage Group.

STGABSYS Automatic backup system name.

STGADMP Automatic dump status.

2.6.2.2 Example 2 - Listing All SMS Data Sets

This command searches all SMS Storage Groups for data sets, and then lists the data sets by dsname and volume.

```
$RSVP STGROUP(*) PRINT(NEW (VOLUME STGROUP STRCLASS -
MGTCLASS DSN)) SMSNMLEN(10)
```

The command generates the following customized report:

```
$RS0A03 RSVP VERSION 1.0     $RSVP  STGROUP(*)                    PAGE 1
VOLUME  STGROUP     STRCLASS    MGTCLASS     DSNAME

SMS601                                       BCDEF01.DFPWED3.SMFDATA
SMS602  SGTEST      SCTEST      STANDARD     KLMNO03.$COPYTP.TEST1
SMS602  SGTEST      SCTEST      STANDARD     LINK.LEE.TSTNODC
SMS602  SGTEST      SCTEST      STANDARD     MNOPQ01.$FM.FMWORK
SMS602  SGTEST                               SMSDS.ASMAST.TABLE.D
SMS602  SGTEST                               SMSDS.ASMAST.TABLE.I
SMS602  SGTEST                               SMSDS.KLMNO03.VSAM.TEST1.DATA
SMS602  SGTEST                               SMSDS.KLMNO03.VSAM.TEST1.INDEX
SMS602  SGTEST                               SMSDS.IXRC.T.D
SMS602  SGTEST                               SMSDS.IXRC.T.I
SMS602  SGTEST      SCTEST                   SMSDS.LEE.CNTL
SMS602  SGTEST      SCTEST      STANDARD     SMSDS.LEE.CNTL
SMS602  SGTEST      SCTEST      STANDARD     SMSDS.LEE.TSTNODC
SMS602  SGTEST      SCTEST      STANDARD     SMSDS.LEE.TSTPS
SMS602  SGTEST      SCTEST      STANDARD     SMSDS.SMS601
SMS602  SGTEST      SCTEST      STANDARD     SMSDS.TEMP.DATDAYSD
SMS602  SGTEST      SCTEST      STANDARD     SMSDS.TEMP.DATSDEL
SMS602  SGTEST      SCTEST      DEL2DAY      SMSDS.TEMP.DSN
SMS602  SGTEST      SCTEST      EXP2DAY      SMSDS.TEMP.EXP2DAY
SMS602  SGTEST      SCTEST                   SMSDS.TEMP.PS
SMS602  SGTEST      SCTEST                   SMSDS.TEST01.ASM
VOLUME  STGROUP     STRCLASS    MGTCLASS     DSNAME
SMS601                                       SYS1.VTOCIX.SMS601
SMS602                                       SYS1.VTOCIX.SMS602
SMS601 SGTEST                                SYS1.VVDS.VSMS601
SMS602 SGTEST                                SYS1.VVDS.VSMS602

  |     |           |          |              |
```

Field Descriptions

VOLUME Volume serial number.

STGROUP Name of the SMS Storage Group name associated with the data set.

STRCLASS Name of the SMS Storage Class associated with the data set.

MGTCLASS Name of the SMS Management Class associated with the data set.

DSNAME Name of the SMS data set.

2.6.2.3 Example 3 - Listing All Data Classes in the SMS Database

This command lists all SMS Data Classes with some of their attributes.

```
$RSVP DATCLASS(*) PRINT(NEW (DATCLASS DATRCORG -
DATLRECL DATSPPRI DATSPSEC DATALLOC -
DATDIBLK DATRECFM DATSHROP))  SMSNMLEN(10)
```

The command generates the following customized report:

```
$RS0A03 RSVP VERSION 1.0     $RSVP   DATCLASS(*)                PAGE 1
DATCLASS   DATR  DATLRE  DATSPPRI  DATSPSEC  DATALLOC  DATDIB  DAT  DATSH

ASMSRC     SEQ       80      5000      5000        80      35  FB   (0,0)
DATAF      SEQ       80      5000      5000        80       0  FB   (0,0)
DATAV      SEQ      255      5000      5000       255       0  VB   (0,0)
DIRECT     RRDS       0       100       100      4096       0       (0,0)
ENTRY      ESDS       0       100       100      4096       0       (0,0)
KEYED      KSDS       0       100       100      4096       0       (0,0)
LINEAR     LDS        0       100       100      4096       0       (0,0)
LISTING    SEQ      137     20000     20000       137       0  VB   (0,0)
LOADLIB    SEQ        0        50        50     23476      62  U    (0,0)
SRCFLIB    SEQ       80      5000      5000        80      62  FB   (0,0)
SRCVLIB    SEQ      255      5000      5000       255      62  VB   (0,0)
TEMPFB80   SEQ       80         0         0         0       0  FB   (0,0)
```

Field Descriptions

DATCLASS Name of the SMS Data Class.

DATR (DATRCORG) Data set organization: SEQ, KSDS, ESDS, RRDS, LDS.

DATLRE (DATLRECL) Logical record length.

DATSPPRI Primary allocation of space units.

DATSPSEC Secondary allocation of space units.

DATALLOC Average number of bytes required per record.

DATDIB DATDIBLK) Number of directory blocks.

DAT (DATRECFM) Record format based on JCL specification: U, V, VS, VB, VBS, F, FS, FB, FBS.

DATSH (DATSHROP) Share options for VSAM data sets.

2.6.2.4 Example 4 - Listing All Management Classes in the SMS Database

This command lists all SMS Management Classes with some of their attributes.

```
$RSVP MGTCLASS(*) PRINT(NEW (MGTCLASS MGTAUTBK MGTBADU -
MGTBKDY MGTBKFQ MGTBKVS MGTEXDAT MGTPELEM   -
MGTBVRD MGTBKNP)) SMSNMLEN(10)
```

The command generates the following customized report:

```
$RS0A03 RSVP VERSION 1.0     $RSVP   MGTCLASS(*)                      PAGE 1
MGTCLASS   MGTAUTB MGTB MGTBKD  MGTBKF MGTBKV  MGTEXDAT   MGTPEL MGTBVR  MGTBKN

DBMIG      NOAUTOB BOTH     30       1      2  NOLIMIT         0      2      60
DBSTNDRD   AUTOBKP BOTH     30       1      2  NOLIMIT         0      2      60
DEL2DAY    AUTOBKP BOTH      1       1      9                  1      1       1
EXP2DAY    AUTOBKP BOTH     30       1      2                  0      1      60
EXTRABAK   AUTOBKP BOTH     30       0      5  NOLIMIT         0      5      60
GDG        NOAUTOB BOTH     30       1      2  NOLIMIT         1      1      60
INTERIM    AUTOBKP BOTH     60       1      2  NOLIMIT         1      2      30
MONTHMIG   AUTOBKP BOTH     60       1      2  NOLIMIT         1      1      30
NOACTION   NOAUTOB BOTH     60       1      2  NOLIMIT         0      1      30
NOMIG      NOAUTOB BOTH     60       0      2  NOLIMIT         0      1      30
STANDARD   AUTOBKP BOTH     30       0      2  NOLIMIT         0      1      60
```

Field Descriptions

MGTCLASS Name of the SMS Management Class.

MGTAUTB (MGTAUTBK) Automatic backup status.

MGTB (MGTBADU) Allow administrator or user backup commands status: NONE, ADM, BOTH.

MGTBKD (MGTBKDY) Number of days to retain backup versions (other than most recent) from the date the backup was created.

MGTBKF (MGTBKFQ) Backup frequency.

MGTBKV (MGTBKVS) Number of backup versions.

MGTEXDAT Expiration date.

MGTPEL (MGTPELEM) Number of GDGs on primary storage.

MGTBVR (MGTBVRD) Number of backup versions to keep for a deleted data set.

MGTBKN (MGTBKNP) Number of days to retain the only backup (most recent) version of a deleted data set.

2.6.2.5 Example 5 - Listing All Storage Classes in the SMS Database

This example lists all SMS Storage Classes with some of their attributes.

```
$RSVP STRCLASS(*) PRINT(NEW (STRCLASS STRDBIAS STRDRESP -
STRSBIAS STRSRESP STRGSP STRAVAIL STRUSER   -
STRTIME)) SMSNMLEN(10)
```

The command generates the following customized report:

```
$RS0A03 RSVP VERSION 1.0     $RSVP    STRCLASS(*)                    PAGE 1
STRCLASS   STRDB  STRDRESP  STRSB STRSRESP  STRGSP     STRA STRUSER  STRTIME

BASE       NONE          0  NONE         0  NOGUARANT  STD  C819405  08:59
CRITICAL   NONE         10  NONE        10  NOGUARANT  CONT USRT001  15:45
DBBASE     NONE          0  NONE         0  GUARANTEE  STD  USRT001  15:44
DBCRIT     READ         10  READ        10  GUARANTEE  CONT USRT001  15:45
DBENHANC   READ         10  READ        10  GUARANTEE  STD  USRT001  15:44
DBFAST     READ          5  READ         5  GUARANTEE  STD  USRT001  15:45
DBLOG      READ          5  WRITE        3  GUARANTEE  STD  USRT001  15:44
ENHANCED   NONE         10  NONE        10  NOGUARANT  STD  USRT001  15:45
FASTREAD   READ          5  READ         5  NOGUARANT  STD  USRT001  15:45
FASTWRIT   WRITE         5  WRITE        5  NOGUARANT  STD  USRT001  15:46
NONSMS     NONE          0  NONE         0  NOGUARANT  STD  NIEDI01  16:40
SCTEST     NONE          0  NONE         0  NOGUARANT  STD  NIEDI01  14:29
***
```

Field Descriptions

STRCLASS Name of the SMS Storage Class.

STRDB (STRDBIAS) Direct bias.

STRDRESP Direct millisecond response.

STRSB (STRSBIAS) Sequential bias.

STRSRESP Sequential millisecond response.

STRGSP Guaranteed space status.

STRA (STRAVAIL) Data availability specification status.

STRUSER User ID of the last user to update the SMS Storage Class.

STRTIME Time of the last update to the SMS Storage Class.

2.6.2.6 Example 6 - Listing SMS Data Classes and Storage Classes

This command searches all volumes beginning with the letters SMS, and lists SMS Data Class and Storage Class fields.

```
$RSVP VOL(SMS) PRINT(NEW (DSN DATCLASS STRCLASS DATRCORG STRDBIAS -
STRSBIAS DATLRECL LRECL DATRECFM VOLUME)) CLUSTER -
HEADING1('1DATA CLASS AND STORAGE CLASS FIELDS FOR SMS DATA SETS')
```

The command generates the following customized report:

```
DATA CLASS AND STORAGE CLASS FIELDS FOR SMS DATA SETS                          PAGE 1
DSNAME                      DATCLASS STRCLASS DATR STRDB STRSB DATLRE LRECL DAT VOLUME

BONVI01.DFPWED3.SMFDATA                                              32767     SMS601
LINK.LEE.TSTNODC                     SCTEST        NONE  NONE            80     SMS602
SMSDS.ASMAST.TABLE                                                    2000     SMS602
SMSDS.IXRC.T                                                           600     SMS602
SMSDS.LEE.CNTL               DATAF   SCTEST   SEQ  NONE  NONE   80       80 FB  SMS602
SMSDS.LEE.TSTBLK             DATAF   SCTEST   SEQ  NONE  NONE   80       80 FB  SMS602
SMSDS.LEE.TSTNODC                    SCTEST        NONE  NONE   80              SMS602
SMSDS.LEE.TSTPS              DATAF   SCTEST   SEQ  NONE  NONE   80       80 FB  SMS602
SMSDS.SMS601                         SCTEST        NONE  NONE   80              SMS601
SMSDS.TEMP.DATDAYSD          TEMPFB80 SCTEST   SEQ  NONE  NONE   80       80 FB  SMS602
SMSDS.TEMP.DAYSDEL                   SCTEST        NONE  NONE   80              SMS602
SMSDS.TEMP.DSN                                                          80     SMS602
SMSDS.TEMP.EXP2DAY                   SCTEST        NONE  NONE            80     SMS602
SMSDS.TEST.PS                        SCTEST        NONE  NONE            80     SMS602
SMSDS.TEST01.ASM             ASMSRC  SCTEST   SEQ  NONE  NONE   80       80 FB  SMS601
SMSDS.TEST02.ASM             ASMSRC  SCTEST   SEQ  NONE  NONE   80       80 FB  SMS601
SYS1.VTOCIX.SMS601                                                     2048     SMS601
SYS1.VTOCIX.SMS602                           SEQ  NONE  NONE    0      2048     SMS602
SYS1.VVDS.VSMS601                                                      4089     SMS601
SYS1.VVDS.VSMS602                                                      4089     SMS602
```

Field Descriptions

DSNAME Data set name.

DATCLASS SMS Data Class name.

STRCLASS SMS Storage Class name.

DATR (DATRCORG) SMS Data Class data set organization. SEQ is sequential.

STRDB (STRDBIAS) SMS Storage Class direct bias.

STRSB (STRSBIAS) SMS Storage Class sequential bias.

DATLRE (DATLRECL) SMS Data Class logical record length.

LRECL Logical record length.

DAT (DATRECFM) SMS Data Class record format based on JCL specification.

VOLUME Volume serial number.

2.6.3 OSCAT Selection Examples

2.6.3.1 Example 1 - Listing Space Information for Cataloged Disk Data Sets

This command searches the catalog for data sets beginning with CAI.AGS410 and prints space information.

```
$RSVP OSCATALOG LEVEL(CAI.AGS410) LIST(SPACE)
```

```
 DSCOUNT DSORG LSTUS       ALLOC     USED   UNUSED   EXT VOLUME DSNAME
          PS   04/27/00      475      475        0     1 SJ0001 CAI.AGS410.ARCH.$ARQUEUE
          PS   11/21/97       47       47        0     1 SJ0001 CAI.AGS410.ARCH.$DUPPOOL
          PS   12/11/97      427      427        0     1 SJ0002 CAI.AGS410.ARCH.$RAQUEUE
          PS   04/14/00      142       47       95     1 SJ0001 CAI.AGS410.ARCH.ARCLOG
          VS   06/10/00      712      712        0     1 SJ0001 CAI.AGS410.ASM2JNL.DATA
          PS   06/04/00      427      427        0     1 SJ0001 CAI.AGS410.BKUP.$RAQUEUE
          PS   02/28/00       47       47        0     1 SJ0001 CAI.AGS410.BKUP.$TAPPOOL
          PS   06/03/00      142      142        0     1 SJ0001 CAI.AGS410.BKUP.ARCLOG
          PS   12/09/97       47       47        0     1 SJ0001 CAI.AGS410.BKUP.ARCPARMS
          PS   02/03/00      237      237        0     1 SJ0001 CAI.AGS410.BKUP.INCR
          PS   02/28/00       47       47        0     1 SJ0005 CAI.AGS410.BKUP.LODSYS
          PS   02/25/00       47       47        0     1 SJ0005 CAI.AGS410.BKUP.LODUSR
          PS   01/31/00       47       47        0     1 SJ0005 CAI.AGS410.BKUP.LOUSER
          PS   06/04/00       47       47        0     1 SJ0005 CAI.AGS410.BKUP.TAPELOG
          PO   06/18/00      665      285      380     1 SJ0001 CAI.AGS410.JCL
      16                    3556     3081      475    15        TOTAL
```

Field Descriptions

DSCOUNT Data set count.

DSORG Data set organization.

LSTUS Last referenced date in Gregorian (MMDDYY).

ALLOC Data set allocation quantity.

USED Quantity of space used.

UNUSED Quantity of unused space.

EXT Number of extents in the data set.

VOLUME Volume serial number.

DSNAME Data set name.

2.6.3.2 Example 2 - Queueing for Archival Data Sets With Specific High-level Node

This command searches the catalog for disk data sets beginning with the prefix CAI and queues them for archive by CA-ASM2. It also produces a report containing the data sets processed and their characteristics.

```
$RSVP OSCAT LEVEL(CAI) SUBCOMMAND($AR) -
PRINT(NEW (DSNAME VOLUME ALLOC USED DSORG)) CLUSTER TRK
```

```
DSNAME                                    VOLUME    ALLOC     USED DSORG                          PAGE     1

CAI.AGS400.ARCH.SCRATCH                   SJ0001       20        1  DA
CAI.AGS400.JCL                            SJ0001       67       57  PO
CAI.AGS400.LOADTEST                       SJ0004       21        1  PO
CAI.AGS410.ASM2IPC                        SJ0005        0        0  VS
CAI.AGS410.ASM2IPC.DATA                   SJ0005       30       30  VS
CAI.AGS410.ASM2IPC.INDEX                  SJ0005       30       30  VS
CAI.AGS410.ASM2JNL                        SJ0005        0        0  VS
CAI.AGS410.ASM2JNL.DATA                   SJ0005      150      150  VS
CAI.AGS410.JCL                            SJ0001      127      125  PO
CAI.AGS410.LOADTEST                       SJ0001       30        4  PO
CAI.ARCH.ARCLOG                           SJ0004        4        4  PS
CAI.ARCH.ARCPARMS                         SJ0001        1        1  PS
CAI.ARCH.DATA                             SJ0004        3        1  PO
CAI.ARCH.INCMODEL                         SJ0001        0        0  PS
CAI.ASM2.CLIST                            SJ0001        5        3  PO
CAI.ASM2.JCL                              SJ0001       41       33  PO
CAI.ASM2IXR.TA58B4C5.D3ECA4D5             SJ0001        0        0  VS
CAI.ASM2IXR.TA58B4C5.D3ECA4D5.DATA        SJ0001       15       15  VS
CAI.ASM2IXR.TA58B4C5.D3ECA4D5.INDEX       SJ0001        1        1  VS
CAI.ASM2L2.ASM                            SJ0004        3        2  P
CAI.ASM241.CONTENTS                       SJ0001        1        1  PS
CAI.A92118.COVER                          SJ0001        1        1  PS
CAI.A92118.SAMPJCL                        SJ0004        3        2  PO
CAI.BKUP.$TAPPOOL                         SJ0001        1        1  PS
CAI.BKUP.LOUSER                           SJ0001        1        1  PS
CAI.CLIST                                 SJ0001       15        1  PO
CAI.IPCSDSET.DIRECTRY                     SJ0001        0        0  VS
CAI.IPCSDSET.DIRECTRY.DATA                SJ0001       15       15  VS
CAI.ISAM                                  SJ0005      105      105  IS
CAI.ISMF@V23.ISPCLIB                      SJ0003       10        5  PO
CAI.ISPF.ISPPROF                          SJ0001       10        6  PO
CAI.ISPFPARM.CLIST                        SJ0001        1        1  PS
CAI.L140.CAIM.CA76696.F1                  SJ0004        3        1  PO
CAI.L140.CAIM.CA76704.F1                  SJ0004        3        1  PO
CAI.L2SMPE.JCL                            SJ0004       60        2  PO
CAI.MOOWO01.BKUP.CATALOG                  SJ0005       12       12  PS
CAI.QC.TST3KSDS.AIX                       SJ0005        0        0  VS
CAI.REPRO.JNL.DATA                        SJ0002      225      225  VS
CAI.SASDATA                               SJ0005        5        0  DA
CAI.SMALL.ASM2.ARCH.CATALOG               SJ0005        1        1  PS
CAI.SP310.IEFZB4D0                        SJ0003        1        1  PO
CAI.SRCHFOR.LIST                          SJ0004        3        1  PS
CAI.STARDETL.REPORTS                      SJ0001        1        1  PS
CAI.SYSPROG.ASM2.ARCH.CATALOG             SJ0005      700      692  PS
CAI.SYSPROG.ASM2.ARCH.CATALOG.NULL        SJ0005        1        0  PS
CAI.SYS2.ASM2.V310.ARCH.LOAD              SJ0005      135      119  PO
CAI.TC58025.OSJCL                         SJ0001        1        1  PS
CAI.TEST.$RS60ICF.CLUSTER                 SJ0005        0        0  VS
CAI.TEST.$RS60ICF.DATA                    SJ0005        1        1  VS
TOTAL                                               1983     1803
```

Field Descriptions

DSNAME	Data set name.
VOLUME	Volume serial number.
ALLOC	Data set allocation quantity.
USED	Quantity of space used.
DSORG	Data set organization.

2.6.4 ASM2CAT Selection Examples (CA-ASM2)

2.6.4.1 Example 1 - Listing All Unloaded Data Sets in the CA-ASM2 Catalog

This command lists all archived and backed up data sets in the IPC.

```
$RSVP ASM2CAT LEVEL(*) IF(IPCTYPE EQ U)  PRINT(NEW (VOLUME -
IPCDATE IPCTIME IPCTYPE SC UNLRETPD IPCUTYPE UNLVOL DSNAME))
```

The command generates the following customized report:

```
$RS0A03 RSVP VERSION 1.0     $RSVP   ASM2CAT LEVEL(*)                              PAGE 1
 VOLUME IPCDA IPCTIME  IPCTYP SC UNLRETPD IPCUTYP UNLVOL DSNAME

 SJ0001 00054 14:13:00    U   AC      730 BACKUP  ASH002 CAI.RMF.FIX.TEST
 SJ0001 00054 14:13:00    U   AC      730 BACKUP  ASH032 CAI.RMF.FIX.TEST
 SJ0001 00024 14:13:00    U   AC      730 BACKUP  ASF070 CAI.RMF.FIX.TEST
 SJ0001 00054 14:13:00    U   AC      730 ARCHIVE ASF062 CAI.RMF.FIX.TEST
 SJ0001 00054 14:13:00    U   AC      730 ARCHIVE ASF062 CAI.RMF.FIX.TEST
 SJ0001 00033 17:34:00    U   AC      730 ARCHIVE ASF060 CAI.RMF.TEST
 SJ0001 00033 17:34:00    U   AC      730 ARCHIVE ASF060 CAI.RMF.TEST
 SJ0001 00037 14:25:00    U   AC      730 ARCHIVE ASF060 CAI.RMF.TEST
 SJ0001 00037 14:25:00    U   AC      730 ARCHIVE ASF060 CAI.RMF.TEST
 SJ0001 00038 17:08:00    U   AC      730 ARCHIVE ASF060 CAI.RMF.TEST
 SJ0001 00038 18:29:00    U   AC      730 ARCHIVE ASF060 CAI.RMF.TEST
 SJ0001 00012 12:45:00    U   AC      730 ARCHIVE ASF062 CAI.$FM.NO.DATA.IN.IXR.CATALOG

   |    |       |      |  |       |    |      |     |    |    |

                                                                                PAGE 2

 VOLUME IPCDA IPCTIME  IPCTYP SC UNLRETPD IPCUTYP UNLVOL DSNAME

 SJ0001 99012 12:45:00    U   AC      730 ARCHIVE ASF062 CAI.ASMR.$FM.TEST3
 SJ0001 98278 13:50:00    U   AC      730 ARCHIVE ASF051 CAI.ASMR.$IXRMMRG.TEST
 SJ0001 98278 13:50:00    U   AC      730 ARCHIVE ASF051 CAI.ASMR.$IXRMMRG.TEST
 SJ0001 99040 18:28:00    U   AC      730 ARCHIVE ASF060 CAI.ASMR.$UL.TEST05
 SJ0001 99040 18:28:00    U   AC      730 ARCHIVE ASF060 CAI.ASMR.$UL.TEST05
 SJ0001 99040 18:28:00    U   AC      730 ARCHIVE ASF060 CAI.ASMR.$UL.TEST06
 SJ0001 99040 18:28:00    U   AC      730 ARCHIVE ASF060 CAI.ASMR.$UL.TEST06
 SJ0001 99040 18:28:00    U   AC      730 ARCHIVE ASF062 CAI.ASMR.$UL.TEST07
 SJ0001 99040 18:28:00    U   AC      730 ARCHIVE ASF062 CAI.ASMR.$UL.TEST07

   |    |       |      |  |       |    |      |     |    |    |

                                                                                PAGE 3
 VOLUME IPCDA IPCTIME  IPCTYP SC UNLRETPD IPCUTYP UNLVOL DSNAME

 SJ0001 99040 18:28:00    U   AC      730 ARCHIVE ASF062 CAITEST.$UL.TEST31
 SJ0001 99040 18:28:00    U   AC      730 ARCHIVE ASF062 CAITEST.$UL.TEST31
 SJ0001 99040 18:28:00    U   AC      730 ARCHIVE ASF062 CAITEST.$UL.TEST32
 SJ0001 99040 18:28:00    U   AC      730 ARCHIVE ASF062 CAITEST.$UL.TEST32

   |    |       |      |  |       |    |      |     |    |    |

                                                                                PAGE 4
 SJ0001 99040 18:28:00    U   AC      730 ARCHIVE ASF062 CAITEST.$UL.TEST33
 SJ0001 99040 18:28:00    U   AC      730 ARCHIVE ASF062 CAITEST.$UL.TEST33

   |    |       |      |  |       |    |      |     |    |    |
```

Field Descriptions

VOLUME	Volume serial number where the data set resided before it was unloaded.
IPCDA	(IPCDATE) Date the unload or $DEFRAG processing occurred.
IPCTIME	Time the unload or $DEFRAG processing occurred.
IPCTYP	(IPCTYPE) IPC record type: D (Defrag volume record) or U (Unload record).
SC	Primary input source: BD (BILLDATA), MS (MSVOLGRP), VO (VOLUMES), OC (OSCATALOG), AC (ASM2CAT), SV (STGROUPS), DC (DATCLASS), MC (MGTCLASS), SC (STRCLASS), SG (STORGACD).
UNLRETPD	(UNLRETPD) Retention period in days for the unloaded data set.
IPCUTYP	(IPCUTYPE) IPC unload record type: ARCHIVE or BACKUP.
UNLVOL	Volume serial to which the data set was unloaded.
DSNAME	Name of the data set.

2.6.4.2 Example 2 - Listing All Data Sets Unloaded to the Disk Staging Area

This command searches the IPC for all data sets unloaded to the Disk Staging Area (DSA).

```
$RSVP ASM2CAT LEV(*)  IF(UNLMDTYP EQ 3390) PRINT(NEW (UNLVOL UNLMDTYP DAY SLFTS -
UNLBYTES UNLCBYTE UNLBLKSZ UNLCOMP UNLCTECH  -
UNLMAMT UNLTRKS UNLODSN))  NOSORT
```

This command generates the following customized report:

```
UNLVOL UNLMDT   DAYSLFTS     UNLBYTES     UNLCBYTE UNLBLKSZ UNLCOMP UNLCTECH      UNLMAMT  UNLTRKS UNLODSN
WRK04A 3390          -41       134720        63068    23476 COMPRES STD             63160        2 ASM2D.B1999325.T123654.A
WRK04A 3390          -41         6592         1195    23476 COMPRES STD              1223        1 ASM2D.B1999325.T123654.A
WRK04A 3390          -41        20720        11044    23476 COMPRES STD             11072        1 ASM2D.B1999325.T123654.A
WRK04A 3390          -41       175122       122131    23476 COMPRES STD            122259        3 ASM2D.B1999325.T123654.A
WRK04A 3390          -41         3152          176    23476 COMPRES STD               192        1 ASM2D.B1999325.T123654.A
WRK04A 3390          -41        11061         6511    23476 COMPRES STD              6539        1 ASM2D.B1999325.T123654.A
WRK04A 3390          -41         3312          230    23476 COMPRES STD               258        1 ASM2D.B1999325.T123654.A
WRK04A 3390          -41         3552          317    23476 COMPRES STD               345        1 ASM2D.B1999325.T123654.A
SJ0005 3390           23         5232         1388    23476 COMPRES STD              1416        1 ASM2D.B2000024.T175041.A
SJ0005 3390           15      1493632       659520    23476 COMPRES STD            660312       15 ASM2D.B1998016.T152436.A
WRK04A 3390          -41         3552          329    23476 COMPRES STD               357        1 ASM2D.B1999325.T123654.A
WRK04A 3390          -50        19112        13485    23476 COMPRES STD             13513        1 ASM2D.B1997316.T124552.A
SJ0005 3390           29        76147         3828    23476 COMPRES STD              3868        1 ASM2D.B1998030.T142225.A
WRK04A 3390          -41       183472       148029    23476 COMPRES STD            148165        4 ASM2D.B1999325.T123654.A
SJ0005 3390            5       241348       149845    23476 COMPRES STD            149981        4 ASM2D.B2000006.T171348.A
SJ0005 3390          -20       241348       149845    23476 COMPRES STD            149981        4 ASM2D.B1999346.T161810.A
SJ0005 3390            5      2061528       184187    23476 COMPRES STD            184871        4 ASM2D.B2000006.T171348.A
SJ0005 3390          -20      2061528       184187    23476 COMPRES STD            184871        4 ASM2D.B1999346.T161810.A
SJ0005 3390            5      2059528       182178    23476 COMPRES STD            182862        4 ASM2D.B1998006.T171348.A
SJ0005 3390          -20      2059528       182178    23476 COMPRES STD            182862        4 ASM2D.B1997346.T161810.A
SJ0005 3390            5       244668       153329    23476 COMPRES STD            153525        4 ASM2D.B1998006.T171348.A
SJ0005 3390          -20       244668       153329    23476 COMPRES STD            153525        4 ASM2D.B1997346.T161810.A
SJ0005 3390            5       241060       149806    23476 COMPRES STD            149942        4 ASM2D.B2000006.T171348.A
SJ0005 3390          -20       241060       149806    23476 COMPRES STD            149942        4 ASM2D.B1999346.T161810.A
SJ0005 3390            5       241188       149859    23476 COMPRES STD            149995        4 ASM2D.B2000006.T171348.A
SJ0005 3390          -20       241188       149859    23476 COMPRES STD            149995        4 ASM2D.B1999346.T161810.A
SJ0005 3390            5       238796       149117    23476 COMPRES STD            149277        4 ASM2D.B1998006.T171348.A
SJ0005 3390          -20       238796       149117    23476 COMPRES STD            149277        4 ASM2D.B1997346.T161810.A
WRK04A 3390          -41        19140         9921    23476 COMPRES STD              9949        1 ASM2D.B1999325.T123654.A
                               28704263     15625215                           15641310
```

Field Descriptions

UNLVOL Identifies the DSA volume.

UNLMDTY Specifies the unload volume media type.

DAYSLFTS Number of days eligible to remain in the Disk Staging Area (DSA). A negative number represents the number of days past the eligibility date.

UNLBYTES Number of bytes unloaded.

UNLCBYTE Byte count after compression.

UNLBLKSZ Physical block size of the unloaded data set on the output media.

UNLCOMP Indicates if the unloaded data set is compressed. The field contains the value COMPRES if the data set is compressed or NOTCOMP if not.

UNLCTECH The data compression technique used to unload the data set.

UNLMAMT Total number of bytes occupied by the data set in the Disk Staging Area. Total represents data bytes after compression + overhead.

UNLTRKS Number of tracks occupied by the unloaded data set in the Disk Staging Area.

UNLODSN Name of the data set on the unload media.

2.6.4.3 Example 3 - Listing All Data Sets Backed Up to the Disk Staging Area

This command searches the IPC for all data sets backed up to the Disk Staging Area (DSA).

```
$RSVP ASM2CAT LEV(*) IF(IPCUTYPE EQ BACKUP) ANDIF1(UNLMDTYP EQ 3390) -
PRINT(NEW (DSNAME IPCUTYPE UNLVOL UNLODSN))
```

This command generates the following customized report:

```
DSNAME                                    IPCUTYP UNLVOL UNLODSN

BCDEF01.TEST.ARCH.LOAD                    BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0003
KLMNO03.ASM2SMON.TESTDATA.DSET0001        BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0005
KLMNO03.ASM2SMON.TESTDATA.DSET0002        BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0009
KLMNO03.ASM2SMON.TESTDATA.DSET0003        BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0011
CDEFG02.MAINTWK                           BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0012
CDEFG02.UN890606.IXRCAT                   BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0027
CN9000.DGS220.LOAD                        BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0007
CN9000.DLA302.ARCH.CATALOG                BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0013
CN9000.DLA302.BKUP.CATALOG                BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0022
C900044.ARCH.ARCPARMS                     BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0025
DLDLV.P991179.ARCH.CAT                    BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0026
JKLMN01.ASM2.MACLIB                       BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0021
JKLMN01.TEST.ASM2IPC                      BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0015
LMNOP01.LINK.LOAD                         BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0018
LNOPQ01.VSAM.ESDS                         BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0004
MOONO01.L2SMPE.PTFIN                      BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0010
MOONO01.REL400.BKUP.ARCLOG                BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0006
MNOPQ02.ISPRINT                           BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0017
MNOPQ02.ISPUNCH                           BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0019
MNOPQ02.RSVP.TABLE                        BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0020
MNOPQ02.SYSIN                             BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0023
NOPQR01.CART.$TAPPOOL                     BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0002
NOPQR01.ISAM                              BACKUP  SJ0005 NOPQR01.ASM2D.B2000024.T175041.A0001
NOPQR01.MISC.CNTL                         BACKUP  SJ0005 NOPQR01.ASM2D.B1998016.T152436.A0001
NOPQR01.NEWNAME.CART.$TAPPOOL             BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0016
NOPQR01.TEMP.PSV                          BACKUP  WRK04A NOPQR01.ASM2D.B1997316.T124552.A0001
NOPQR01.TEST.LARGE                        BACKUP  SJ0005 NOPQR01.ASM2D.B1998030.T142225.A0001
NOPQR01.TEST.LOAD                         BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0024
NOPQR01.TESTDATA.PO.VB                    BACKUP  SJ0005 NOPQR01.ASM2D.B1997346.T161810.A0001
NOPQR01.TESTDATA.PO.VB                    BACKUP  SJ0005 NOPQR01.ASM2D.B1998006.T171348.A0001
NOPQR01.TESTDATA.PS.F                     BACKUP  SJ0005 NOPQR01.ASM2D.B1997346.T161810.A0002
NOPQR01.TESTDATA.PS.F                     BACKUP  SJ0005 NOPQR01.ASM2D.B1998006.T171348.A0002
NOPQR01.TESTDATA.PS.FB                    BACKUP  SJ0005 NOPQR01.ASM2D.B1997346.T161810.A0003
NOPQR01.TESTDATA.PS.FB                    BACKUP  SJ0005 NOPQR01.ASM2D.B1998006.T171348.A0003
NOPQR01.TESTDATA.PS.U                     BACKUP  SJ0005 NOPQR01.ASM2D.B1997346.T161810.A0004
NOPQR01.TESTDATA.PS.U                     BACKUP  SJ0005 NOPQR01.ASM2D.B1998006.T171348.A0004
NOPQR01.TESTDATA.PS.VB                    BACKUP  SJ0005 NOPQR01.ASM2D.B1997346.T161810.A0005
NOPQR01.TESTDATA.PS.VB                    BACKUP  SJ0005 NOPQR01.ASM2D.B1998006.T171348.A0005
NOPQR01.TESTDATA.PS.VBS                   BACKUP  SJ0005 NOPQR01.ASM2D.B1997346.T161810.A0006
NOPQR01.TESTDATA.PS.VBS                   BACKUP  SJ0005 NOPQR01.ASM2D.B1998006.T171348.A0006
NOPQR01.TESTDATA.VS                       BACKUP  SJ0005 NOPQR01.ASM2D.B1997346.T161810.A0007
NOPQR01.TESTDATA.VS                       BACKUP  SJ0005 NOPQR01.ASM2D.B1998006.T171348.A0007
VTOC1.VSJ0004                             BACKUP  WRK04A NOPQR01.ASM2D.B1997325.T123654.A0001
```

Field Descriptions

DSNAME Data set name.

IPCUTYPE IPC unload record type.

UNLVOL Volume serial to which the data set was unloaded. In this example, the DSA volume.

UNLODSN Data set name on the unload media. In this example, data set name on DSA.

2.6.4.4 Example 4 - Listing All Data Sets Unloaded to a Specific DSA Volume

This command searches the IPC for all data sets unloaded to DSA volume WRK04A.

```
$RSVP ASM2CAT LEV(*) IF(UNLVOL EQ WRK04A) PRINT(NEW (DSNAME -
VOLUME UNLODSN))
```

This command generates the following customized report:

```
DSNAME                                  VOLUME UNLODSN

BOXER01.TEST.ARCH.LOAD                  SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0003
KLMNO03.ASM2SMON.TESTDATA.DSET0001      SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0005
KLMNO03.ASM2SMON.TESTDATA.DSET0002      SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0009
KLMNO03.ASM2SMON.TESTDATA.DSET0003      SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0011
CDEFG02.MAINTWK                         SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0012
CDEFG02.UN890606.IXRCAT                 SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0027
CN9000.DGS220.LOAD                      SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0007
CN9000.DLA302.ARCH.CATALOG              SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0013
CN9000.DLA302.BKUP.CATALOG              SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0022
C900044.ARCH.ARCPARMS                   SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0025
DLDLV.P991179.ARCH.CAT                  SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0026
JKLMN01.ASM2.MACLIB                     SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0021
JKLMN01.TEST.ASM2IPC                    SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0015
LEXXY01.LINK.LOAD                       SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0018
LMNOP01.VSAM.ESDS                       SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0004
MOONY01.L2SMPE.PTFIN                    SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0010
MOONY01.REL400.BKUP.ARCLOG              SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0006
MNOXY02.ISPRINT                         SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0017
MNOXY02.ISPUNCH                         SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0019
MNOXY02.RSVP.TABLE                      SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0020
MNOXY02.SYSIN                           SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0023
NOPQR01.CART.$TAPPOOL                   SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0002
NOPQR01.NEWNAME.CART.$TAPPOOL           SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0016
NOPQR01.TEMP.PSV                        SJ0005 NOPQR01.ASM2D.B1997316.T124552.A0001
NOPQR01.TEST.LOAD                       SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0024
VTOC1.VSJ0004                           SJ0004 NOPQR01.ASM2D.B1997325.T123654.A0001
```

Field Descriptions

DSNAME Data set name.

VOLUME The volume on which the unloaded data set resided.

UNLODSN Data set name on the unload media. In this example, data set name on DSA.

2.6.5 TMC Selection Examples (CA-1)

2.6.5.1 Example 1 - Searching for Active Data Sets Created on a Specific Drive

```
$RSVP CA1TMC IF(TLUNIT EQ E80) AND1(TOOUTIN EQ Y) AND2(TSCRIND EQ N) -
PRINT(NEW (TVOLSER TFILSEQ TLUNIT TLDATE TLTIME TLJOB DSNAME))
```

This command lists all active data sets in the TMC created on tape drive E80:

```
TVOLSER TFILSEQ TLUNIT TLDATE     TLTIME   TLJOB    DSNAME
 980021      1    E80  1999/066   14:21:00 MNOPQ01V MNOPQ01.VOLUME1.FILE1
 980020      1    E80  1999/066   14:25:00 MNOPQ01V MNOPQ01.VOLUME1.FILE1
 980020      2    E80  1999/066   14:28:00 MNOPQ01V MNOPQ01.VOLUME1.FILE2
```

Field Descriptions

TVOLSER Tape volume serial number.

TFILSEQ Tape file sequence number.

TLUNIT Tape use unit address.

TLDATE Tape last use date.

TLTIME Tape last use time.

TLJOB Tape last use job name.

DSNAME Data set name.

2.6.5.2 Example 2 - Searching the CA1 TMC for Exceptional Conditions

```
$RSVP CA1TMC IF(TTPMIND EQ Y) AND1(TOOUTIN EQ Y ) AND2(TSCRIND NE Y) -
PRINT(NEW (TVOLSER TFILSEQ TTPMIND TOOUTIN TSCRIND DSNAME))
```

This command lists all temporary files in the TMC which are in nonscratch status:

```
TVOLSER TFILSEQ TTPMIND TOOUTIN TSCRIND DSNAME
 980001      1     Y       Y       N    SYS95058.T125159.RA000.WXYZZ01B.R0000001
```

Field Descriptions

TVOLSER Tape volume serial number.

TFILSEQ Tape file sequence number.

TTPMIND Tape temporary data set indicator.

TOOUTIN Tape opened for output indicator.

TSCRIND Tape common scratch indicator.

DSNAME Data set name.

2.6.5.3 Example 3 - Listing All Tape Volumes in the TMC Eligible for Reuse

```
$RS CA1TMC IF(TSCRIND EQ Y) AND1(TDELIND EQ N) -
PRINT(NEW (TVOLSER TSCRIND TDELIND TUSETOT TDTECLN))
```

This command lists all volumes in the TMC which are in scratch/nondelete status:

```
TVOLSER TSCRIND TDELIND TUSETOT TDTECLN
 980000    Y       N          0 1990/001
 980001    Y       N          1 ZEROS
 980005    Y       N          0 ZEROS
 980008    Y       N         31 1997/058
 980009    Y       N          4 ZEROS
 980010    Y       N          4 ZEROS
 980024    Y       N          4 ZEROS
 980025    Y       N          4 ZEROS
 980026    Y       N          4 ZEROS
 980027    Y       N          4 ZEROS
 980028    Y       N          4 ZEROS
 980029    Y       N          4 ZEROS
 980030    Y       N          4 ZEROS
 980031    Y       N          4 ZEROS
 980032    Y       N          4 ZEROS
 980033    Y       N          4 ZEROS
 980034    Y       N          4 ZEROS
 980035    Y       N          4 ZEROS
 980036    Y       N          0 ZEROS
 980037    Y       N          0 ZEROS
 980038    Y       N          0 ZEROS
 980039    Y       N          0 ZEROS
 980040    Y       N          0 ZEROS
```

Field Descriptions

TVOLSER Tape volume serial number.

TSCRIND Tape common scratch indicator.

TDELIND Tape delete indicator.

TUSETOT Tape use totals.

TDTECLN Tape date cleaned.

2.6.5.4 Example 4 - Listing All Files on Tapes Whose Volser Starts with TST03

```
$RSVP CA1TMC -
IF(TVOLSER EQ TST03) -
SORT(TVOLSER) -
PRINT(NEW (TVOLSER TFILSEQ TSCRIND DSNAME)) -
HEADING1('1  LISTING OF ALL TAPES WITH VOLSERS STARTING WITH TST03')
```

Only files on tapes whose volser starts with the character string TST03 are reported. The sort order is by tape volser.

The set of elements that are printed is completely specified in this example by the PRINT(NEW (---)) clause:

```
          LISTING OF ALL TAPES WITH VOLSERS STARTING WITH TST03                                    PAGE     1

TVOLSER TFILSEQ TSCRIND DSNAME

 TST030      2     N    SYS92100.T065453.RA000.MALCH01C.R0000001
 TST030      1     N    MNOPQ01.COPYCAT.ONE.FILE
 TST031      1     Y    MNOPQ01.COPYCAT.MVOL.FILE
 TST032      1     Y    MNOPQ01.COPYCAT.MVOL.FILE
 TST033      1     Y    MNOPQ01.COPYCAT.MVOL.FILE
 TST034      5     N    MNOPQ01.COPYCAT.MDS.FILE5
 TST034      4     N    MNOPQ01.COPYCAT.MDS.FILE4
 TST034      3     N    MNOPQ01.COPYCAT.MDS.FILE3
 TST034      2     N    MNOPQ01.COPYCAT.MDS.FILE2
 TST034      1     N    MNOPQ01.COPYCAT.MDS.FILE1
 TST035      3     N    MNOPQ01.COPYCAT.MDSMVOL.FILE3
 TST035      2     N    MNOPQ01.COPYCAT.MDSMVOL.FILE2
 TST035      1     N    MNOPQ01.COPYCAT.MDSMVOL.FILE1
 TST036      3     N    MNOPQ01.COPYCAT.MDSMVOL.FILE3
 TST037      5     N    MNOPQ01.COPYCAT.MDSMVOL.FILE5
 TST037      4     N    MNOPQ01.COPYCAT.MDSMVOL.FILE4
 TST037      3     N    MNOPQ01.COPYCAT.MDSMVOL.FILE3
```

Field Descriptions

TVOLSER Tape volume serial number.

TFILSEQ Tape file sequence number.

TSCRIND Tape common scratch indicator.

DSNAME Data set name.

2.6.5.5 Example 5 - Listing Files by Tape Volser and File Sequence Number

```
$RSVP CA1TMC SORT(TVOLSER TFILSEQ)            -
IF(TVOLSER GT TST03-)                        -
AND1(TVOLSER LT TST05)                       -
PRINT(NEW (TVOLSER TFILSEQ TSCRIND DSNAME))   -
HEADING1('1  LISTING OF TAPES WITH VOLSERS STARTING TST03 OR TST04')
```

Only files on tapes whose volser starts with the character string TST03 or TST04 are reported. The sort order is by tape volser and file sequence on the tape.

The set of elements that are printed is completely specified in this example by the PRINT(NEW (---)) clause:

```
          LISTING OF TAPES WITH VOLSERS STARTING TST03 OR TST04                                    PAGE     1

TVOLSER TFILSEQ TSCRIND DSNAME

 TST030      1     N    MNOPQ01.COPYCAT.ONE.FILE
 TST030      2     N    SYS92100.T065453.RA000.MALCH01C.R0000001
 TST031      1     Y    MNOPQ01.COPYCAT.MVOL.FILE
 TST032      1     Y    MNOPQ01.COPYCAT.MVOL.FILE
 TST033      1     Y    MNOPQ01.COPYCAT.MVOL.FILE
 TST034      1     N    MNOPQ01.COPYCAT.MDS.FILE1
 TST034      2     N    MNOPQ01.COPYCAT.MDS.FILE2
 TST034      3     N    MNOPQ01.COPYCAT.MDS.FILE3
 TST034      4     N    MNOPQ01.COPYCAT.MDS.FILE4
 TST034      5     N    MNOPQ01.COPYCAT.MDS.FILE5
 TST035      1     N    MNOPQ01.COPYCAT.MDSMVOL.FILE1
 TST035      2     N    MNOPQ01.COPYCAT.MDSMVOL.FILE2
 TST035      3     N    MNOPQ01.COPYCAT.MDSMVOL.FILE3
 TST036      3     N    MNOPQ01.COPYCAT.MDSMVOL.FILE3
 TST037      3     N    MNOPQ01.COPYCAT.MDSMVOL.FILE3
 TST037      4     N    MNOPQ01.COPYCAT.MDSMVOL.FILE4
 TST037      5     N    MNOPQ01.COPYCAT.MDSMVOL.FILE5
 TST041      1     N    MNOPQ01.COPYCAT.MDS.FILE1
 TST041      2     N    MNOPQ01.COPYCAT.MDS.FILE2
 TST041      3     N    MNOPQ01.COPYCAT.MDS.FILE3
 TST041      4     N    MNOPQ01.COPYCAT.MDS.FILE4
 TST041      5     N    MNOPQ01.COPYCAT.MDS.FILE5
 TST042      1     N    MNOPQ01.COPYCAT.MDS.FILE1
 TST042      2     N    MNOPQ01.COPYCAT.MDS.FILE2
 TST042      3     N    MNOPQ01.COPYCAT.MDS.FILE3
 TST042      4     N    MNOPQ01.COPYCAT.MDS.FILE4
 TST042      5     N    MNOPQ01.COPYCAT.MDS.FILE5
 TST043      1     N    MNOPQ01.COPYCAT.MDS.FILE1
 TST043      2     N    MNOPQ01.COPYCAT.MDS.FILE2
 TST043      3     N    MNOPQ01.COPYCAT.MDS.FILE3
 TST043      4     N    MNOPQ01.COPYCAT.MDS.FILE4
 TST043      5     N    MNOPQ01.COPYCAT.MDS.FILE5
```

Field Descriptions

TVOLSER Tape volume serial number.

TFILSEQ Tape file sequence number.

TSCRIND Tape common scratch indicator.

DSNAME Data set name.

2.6.6 VMF Selection Examples (CA-Dynam/TLMS)

2.6.6.1 Example 1 - Listing All Volumes in the VMF with the Scratch Indicator Set

```
$RSVP TLMSVMF IF(TSCRIND EQ Y) -
PRINT(NEW (TVOLSER TSCRIND TDATSCR TSCRSOR TACTFIL TOUTLOC))  -
HEADING1('1     SEARCH THE VMF FOR VOLUMES WITH THE SCRATCH INDICATOR -
SET ')
```

This command lists all volumes in the TLMS Volume Master File which have the scratch indicator set.

```
     SEARCH THE VMF FOR VOLUMES WITH THE SCRATCH INDICATOR SET                                    PAGE     1
                                                                                                 19:06:56 MM/DD/YY
TVOLSER TSCRIND TDATSCR    TSCRSOR TACTFIL TOUTLOC
 000023    Y    10/02/2000    1         0    DC
 000024    Y    10/02/2000    1         0    DC
 000025    Y    10/02/2000    1         0    DC
 000034    Y    02/27/2000    3         0    DC
 000040    Y    10/16/2000    2         0    DC
 000044    Y    02/27/2000    3         0    DC
 000048    Y    10/02/2000    1         0    DC
 000058    Y    10/02/2000    1         0    DC
 000059    Y    10/02/2000    1         0    DC
 000060    Y    10/02/2000    1         0    DC
 000061    Y    10/02/2000    1         0    DC
 000064    Y    10/02/2000    1         0    DC
 000071    Y    10/02/2000    1         0    DC
 000072    Y    10/02/2000    1         0    DC
 000073    Y    10/02/2000    1         0    DC
 000122    Y    04/01/2000    3         0    DC
 000150    Y    04/09/2000    3         0    DC
 000153    Y    03/26/2000    3         0    DC
 000200    Y    10/03/2000    1         0    DC
 000202    Y    10/03/2000    1         0    DC
 000203    Y    10/03/2000    1         0    DC
 000205    Y    10/03/2000    1         0    DC
 000300    Y    10/03/2000    1         0    DC
 000301    Y    10/03/2000    1         0    DC
 000302    Y    10/03/2000    1         0    DC
```

```
A41005    Y    10/03/2000    1         0    DC
A41006    Y    10/03/2000    1         0    DC
A41007    Y    10/03/2000    1         0    DC
A41008    Y    10/03/2000    1         0    DC
A41009    Y    10/03/2000    1         0    DC
A1A002    Y    10/03/2000    1         0    DC
A1A003    Y    10/03/2000    1         0    DC
A1A004    Y    10/03/2000    1         0    DC
A1A005    Y    10/03/2000    1         0    DC
A1A006    Y    10/03/2000    1         0    DC
A1A007    Y    10/03/2000    1         0    DC
A1A008    Y    10/03/2000    1         0    DC
A1A009    Y    10/03/2000    1         0    DC
BK0001    Y    10/03/2000    1         0    DC
BK0002    Y    10/03/1997    1         0    DC
BK0003    Y    10/03/1997    1         0    DC
BK0004    Y    10/03/1997    1         0    DC
BK0005    Y    10/03/1997    1         0    DC
BK0006    Y    10/03/1997    1         0    DC
BK0007    Y    10/03/1997    1         0    DC
BK0008    Y    10/03/1997    1         0    DC
BK0009    Y    10/03/1997    1         0    DC
BK0012    Y    10/03/1997    1         0    DC
BK0013    Y    10/03/1997    1         0    DC
BK0014    Y    10/03/1997    1         0    DC
TST002    Y    05/07/1997    3         0    DC
TST006    Y    02/11/1997    3         0    DC
TST031    Y    04/15/1997    3         0    DC
TST032    Y    04/15/1997    3         0    DC
```

Field Descriptions

TVOLSER Tape volume serial number.

TSCRIND Tape common scratch indicator.

TDATSCR Tape date scratched.

TSCRSOR Tape scratch source.

TACTFIL Tape active files on volume.

TOUTLOC Tape output location code.

2.6.6.2 Example 2 - Searching the VMF Using the Selection Keyword CONTAIN

```
$RSVP TLMSVMF CONTAIN(COPYCAT) -
PRINT(NEW (TVOLSER TFILSEQ DSNAME))  -
HEADING1('1     SEARCH THE VMF FOR DATA SETS CONTAINING THE CHARACTERS -
COPYCAT ANYWHERE')  -
HEADING2(' IN THE DATA SET NAME AND PRINT ITS TAPE VOL SER AND FILE SEQ')
```

This command lists all data sets in the TLMS Volume Master File which have COPYCAT anywhere in the data set name.

```
     SEARCH THE VMF FOR DATA SETS CONTAINING THE CHARACTERS COPYCAT ANYWHERE                    PAGE     1
IN THE DATA SET NAME AND PRINT ITS TAPE VOL SER AND FILE SEQ                          19:36:40 MM/DD/YYYY

TVOLSER TFILSEQ DSNAME

 TST011      1  MNOPQ01.COPYCAT.ONE.FILE
 TST012      1  MNOPQ01.COPYCAT.MDS.FILE1
 TST012      2  MNOPQ01.COPYCAT.MDS.FILE2
 TST012      3  MNOPQ01.COPYCAT.MDS.FILE3
 TST012      4  MNOPQ01.COPYCAT.MDS.FILE4
 TST012      5  MNOPQ01.COPYCAT.MDS.FILE5
 TST013      1  MNOPQ01.COPYCAT.MVOL.FILE
 TST014      1  MNOPQ01.COPYCAT.MVOL.FILE
 TST015      1  MNOPQ01.COPYCAT.MVOL.FILE
 TST016      1  MNOPQ01.COPYCAT.MDSMVOL.FILE1
 TST016      2  MNOPQ01.COPYCAT.MDSMVOL.FILE2
 TST016      3  MNOPQ01.COPYCAT.MDSMVOL.FILE3
 TST017      3  MNOPQ01.COPYCAT.MDSMVOL.FILE3
 TST018      3  MNOPQ01.COPYCAT.MDSMVOL.FILE3
 TST018      4  MNOPQ01.COPYCAT.MDSMVOL.FILE4
 TST018      5  MNOPQ01.COPYCAT.MDSMVOL.FILE5
 TST019      1  MNOPQ01.COPYCAT.MDS.FILE1
 TST019      2  MNOPQ01.COPYCAT.MDS.FILE2
 TST019      3  MNOPQ01.COPYCAT.MDS.FILE3
 TST019      4  MNOPQ01.COPYCAT.MDS.FILE4
 TST019      5  MNOPQ01.COPYCAT.MDS.FILE5
 TST020      1  MNOPQ01.COPYCAT.MDS.FILE1
 TST020      2  MNOPQ01.COPYCAT.MDS.FILE2
 TST020      3  MNOPQ01.COPYCAT.MDS.FILE3
 TST020      4  MNOPQ01.COPYCAT.MDS.FILE4
 TST020      5  MNOPQ01.COPYCAT.MDS.FILE5
```

```
TST021      1  MNOPQ01.COPYCAT.MDSMVOL.FILE1
TST021      2  MNOPQ01.COPYCAT.MDSMVOL.FILE2
TST026      1  MNOPQ01.COPYCAT.MDS.FILE1
TST026      2  MNOPQ01.COPYCAT.MDS.FILE2
TST026      3  MNOPQ01.COPYCAT.MDS.FILE3
TST026      4  MNOPQ01.COPYCAT.MDS.FILE4
TST026      5  MNOPQ01.COPYCAT.MDS.FILE5
TST027      1  MNOPQ01.COPYCAT.MVOL.FILE
TST028      1  MNOPQ01.COPYCAT.MVOL.FILE
TST029      1  MNOPQ01.COPYCAT.MVOL.FILE
TST030      1  MNOPQ01.COPYCAT.ONE.FILE
TST031      1  MNOPQ01.COPYCAT.MVOL.FILE
TST032      1  MNOPQ01.COPYCAT.MVOL.FILE
TST033      1  MNOPQ01.COPYCAT.MVOL.FILE
TST034      1  MNOPQ01.COPYCAT.MDS.FILE1
TST034      2  MNOPQ01.COPYCAT.MDS.FILE2
TST034      3  MNOPQ01.COPYCAT.MDS.FILE3
TST034      4  MNOPQ01.COPYCAT.MDS.FILE4
TST034      5  MNOPQ01.COPYCAT.MDS.FILE5
TST035      1  MNOPQ01.COPYCAT.MDSMVOL.FILE1
TST035      2  MNOPQ01.COPYCAT.MDSMVOL.FILE2
TST035      3  MNOPQ01.COPYCAT.MDSMVOL.FILE3
TST036      3  MNOPQ01.COPYCAT.MDSMVOL.FILE3
TST037      3  MNOPQ01.COPYCAT.MDSMVOL.FILE3
TST037      4  MNOPQ01.COPYCAT.MDSMVOL.FILE4
TST037      5  MNOPQ01.COPYCAT.MDSMVOL.FILE5
TST041      1  MNOPQ01.COPYCAT.MDS.FILE1
TST041      2  MNOPQ01.COPYCAT.MDS.FILE2
```

Field Descriptions

TVOLSER Tape volume serial number.

TFILSEQ Tape file sequence number.

DSNAME Data set name.

2.6.6.3 Example 3 - Listing Active Tape Data Sets Created by a Program

```
$RS TLMSVMF IF(TCPGM EQ GREATEST) AND1(TSCRIND NE Y) -
PRINT(NEW (TVOLSER TFILSEQ TSCRIND TCJOB TCPGM TCDATE)) -
HEADING1('1     SEARCH THE VMF FOR ACTIVE DATA SETS CREATED BY PROGRAM  -
GREATEST')
```

This command lists all active data sets created by program GREATEST:

```
     SEARCH THE VMF FOR ACTIVE DATA SETS CREATED BY PROGRAM GREATEST                                    PAGE     1

TVOLSER TFILSEQ TSCRIND TCJOB    TCPGM    TCDATE

 000035      1     N    AA1J401  GREATEST 10/01/2000
 000036      1     N    AA1J401  GREATEST 10/01/2000
 000037      1     N    AA1J401  GREATEST 10/01/2000
 000038      1     N    AA1J401  GREATEST 10/01/2000
 000039      1     N    AA1J401  GREATEST 10/01/2000
 000046      1     N    AA1J401  GREATEST 10/01/2000
 000047      1     N    AA1J402  GREATEST 10/01/2000
 000050      1     N    AA1J402  GREATEST 10/01/2000
 000051      1     N    AA1J402  GREATEST 10/01/2000
 000052      1     N    AA1J402  GREATEST 10/01/2000
 000053      1     N    AA1J402  GREATEST 10/01/2000
 000054      1     N    AA1J402  GREATEST 10/01/2000
 000055      1     N    AA1J402  GREATEST 10/01/2000
```

Field Descriptions

TVOLSER Tape volume serial number.

TFILSEQ Tape file sequence number.

TSCRIND Tape common scratch indicator.

TCJOB Tape creation job.

TCPGM Tape creation program name.

TCDATE Tape creation date.

2.6.6.4 Example 4 - Using the IF/ANDIF Keywords to Search the VMF

```
$RS TLMSVMF IF(TSCRIND EQ Y) ANDIF1(TSCRSOR EQ 3)  -
PRINT(NEW (TVOLSER TSCRIND TDATSCR TSCRSOR))  -
HEADING1('1     TAPE VOLUMES IN THE VMF IN SCRATCH STATUS WITH SCRATCH  -
SOURCE OF 3 ')
```

This command lists all volumes in the TLMS Volume Master File which have the scratch indicator set and have a scratch source of 3:

```
     TAPE VOLUMES IN THE VMF IN SCRATCH STATUS WITH SCRATCH SOURCE OF 3                    PAGE     1
                                                                                 20:38:53 MM/DD/YYYY
TVOLSER TSCRIND TDATSCR    TSCRSOR

 000034    Y    02/27/2000    3
 000044    Y    02/27/2000    3
 000122    Y    04/01/2000    3
 000150    Y    04/09/2000    3
 000153    Y    03/26/2000    3
 TST002    Y    05/07/2000    3
 TST006    Y    02/11/2000    3
 TST031    Y    04/15/2000    3
 TST032    Y    04/15/2000    3
 TST033    Y    04/15/2000    3
```

Field Descriptions

TVOLSER Tape volume serial number.

TSCRIND Tape common scratch indicator.

TDATSCR Tape date scratched.

TSCRSOR Tape scratch source.

2.6.6.5 Example 5 - Listing All Tapes Used for Payroll Showing Their Scratch Status

```
RSVP TLMSVMF LIKE(-.PAYROLL.-)   -
PRINT(NEW (TVOLSER TFILSEQ TSCRIND DSNAME))   -
HEADING1('1  TAPES USED BY PAYROLL SYSTEM REPORT')
```

Only files with PAYROLL somewhere in their data set name are selected for reporting by the LIKE masking facility. By default the report is sorted on DSNAME.

The set of elements that are printed is completely specified in this example by the PRINT(NEW (---)) clause:

```
                        TAPES USED BY PAYROLL SYSTEM REPORT                                    PAGE     1

TVOLSER TFILSEQ TSCRIND DSNAME

 000200      1     Y    DAILY.PAYROLL.JOURNAL
 000203      1     Y    DAILY.PAYROLL.JOURNAL
 000204      1     N    MONTHLY.PAYROLL
 000201      1     N    MONTHLY.PAYROLL
 000205      1     Y    WEEKLY.PAYROLL
 000202      1     Y    WEEKLY.PAYROLL
```

Field Descriptions

TVOLSER Tape volume serial number.

TFILSEQ Tape file sequence number.

TSCRIND Tape common scratch indicator.

DSNAME Data set name.

2.6.6.6 Example 6 - Listing All Tapes Not in Scratch Status

```
$RSVP TLMSVMF LIKE(-BK.-)   -
IF(TFILSEQ EQ 1) AND1(TSCRIND NE Y)   -
SORT(TVOLSER)   -
PRINT(NEW (TVOLSER TFILSEQ DSNAME))   -
HEADING1('1  DASD BACKUP TAPES REPORT')
```

Only files with the characters BK as the last two characters of the high-level qualifier are selected by the LIKE mask. The report is sorted on the tape volser.

The set of elements that are printed is completely specified in this example by the PRINT(NEW (---)) clause.

```
                              DASD BACKUP TAPES REPORT                                          PAGE     1

TVOLSER TFILSEQ DSNAME
 ARC001      1  ASMBK.MANY.FILE1
 ARC003      1  ASMBK.MANY.FILE4
 ARC005      1  ASMBK.MANY.FILE4
 ARC006      1  ASMBK.MANY.FILE4
 ARC008      1  ASMBK.MANY.FILE4
 ARC009      1  ASMBK.MANY.FILE3
 ARC024      1  ASMBK.ANY.FILE
 ARC030      1  ASMBK.MANY.FILE1
```

Field Descriptions

TFILSEQ Tape file sequence number.

DSNAME Data set name.

2.6.7 Combined Selection Examples

2.6.7.1 Example 1 - Searching More Than One Primary Input Source

This command searches three input sources in a single pass. It searches DASD volumes SMS601, SMS602, and SJ0001, the OS catalog, and the IPC for data sets belonging to VWXYZ03.

```
$RSVP VOL(SMS601 SMS602 SJ0001) OSCAT ASM2CAT LEVEL(VWXYZ03) -
PRINT(NEW (IPCDATE IPCTIME IPCTYPE SC UNLRETPD UNLVOL -
STRCLASS MGTCLASS VOLUME DSNAME DATCLASS))
```

The command generates the following customized report:

```
$RS0A03 RSVP VERSION 1.0     $RSVP   VOL  OSCAT  ASM2CAT                              PAGE 1
 IPCDA IPCTIME  IPCTYP SC UNLRETPD UNLVOL STRCLASS  MGTCLASS   VOLUME DSNAME

 00322 17:35:00    U   AC      730 ASF070 SCTEST    STANDARD   SJ0001 VWXYZ03.$COPYTP.TEST1
 00322 17:35:00    U   AC      730 ASF061                      SJ0001 VWXYZ03.$COPYTP.TEST1
 00322 17:35:00    U   AC      730 ASH002                      SJ0001 VWXYZ03.$COPYTP.TEST1
 00322 17:35:00    U   AC      730 ASH032                      SJ0001 VWXYZ03.$COPYTP.TEST1
 00322 17:35:00    U   AC      730 ASF070                      SJ0001 VWXYZ03.$COPYTP.TEST1
                       OC              SCTEST    STANDARD   SMS602 VWXYZ03.$COPYTP.TEST1
                       VO              SCTEST    STANDARD   SMS602 VWXYZ03.$COPYTP.TEST1
 00322 17:35:00    U   AC      730 ASF061                      SJ0001 VWXYZ03.$COPYTP.TEST2
 00322 17:35:00    U   AC      730 ASF061                      SJ0001 VWXYZ03.$COPYTP.TEST2
 00012 12:45:00    U   AC      730 ASF062                      SJ0001 VWXYZ03.$FM.NO.DATA.IN
 00012 12:45:00    U   AC      730 ASF062                      SJ0001 VWXYZ03.$FM.NO.DATA.IN
 00012 12:45:00    U   AC      730 ASF062                      SJ0001 VWXYZ03.$FM.SPANNED.TE
 00012 12:45:00    U   AC      730 ASF062                      SJ0001 VWXYZ03.$FM.SPANNED.TE
 00040 14:47:00    U   AC      730 ASF060                      SJ0001 VWXYZ03.$FM.TEST01
 00040 14:47:00    U   AC      730 ASF060                      SJ0001 VWXYZ03.$FM.TEST01
 00040 14:47:00    U   AC      730 ASF060                      SJ0001 VWXYZ03.$FM.TEST02
 00040 14:47:00    U   AC      730 ASF060                      SJ0001 VWXYZ03.$FM.TEST02
 00040 14:47:00    U   AC      730 ASF060                      SJ0001 VWXYZ03.$FM.TEST03
 00040 14:47:00    U   AC      730 ASF060                      SJ0001 VWXYZ03.$FM.TEST03
 00040 14:47:00    U   AC      730 ASF060                      SJ0001 VWXYZ03.$FM.TEST04
 00040 14:47:00    U   AC      730 ASF060                      SJ0001 VWXYZ03.$FM.TEST04
 00040 14:47:00    U   AC      730 ASF060                      SJ0001 VWXYZ03.$FM.TEST05
 00040 14:47:00    U   AC      730 ASF060                      SJ0001 VWXYZ03.$FM.TEST05
 00040 14:47:00    U   AC      730 ASF060                      SJ0001 VWXYZ03.$FM.TEST06
 00040 14:47:00    U   AC      730 ASF060                      SJ0001 VWXYZ03.$FM.TEST06
 00012 12:45:00    U   AC      730 ASF062                      SJ0001 VWXYZ03.$FM.TEST2
 00012 12:45:00    U   AC      730 ASF062                      SJ0001 VWXYZ03.$FM.TEST2
 00012 12:45:00    U   AC      730 ASF062                      SJ0001 VWXYZ03.$FM.TEST3
 00012 12:45:00    U   AC      730 ASF062                      SJ0001 VWXYZ03.$FM.TEST3
                       VO                                      SJ0001 VWXYZ03.$FM.TEST3
 00040 18:28:00    U   AC      730 ASF062                      SJ0001 VWXYZ03.$UL.TEST16

    |             |   |      |    |                   |            |      |       |

                                                                                 PAGE 2
 IPCDA IPCTIME  IPCTYP SC UNLRETPD UNLVOL STRCLASS  MGTCLASS   VOLUME DSNAME

 00040 18:28:00    U   AC      730 ASF062                      SJ0001 VWXYZ03.$UL.TEST16
 00040 18:28:00    U   AC      730 ASF062                      SJ0001 VWXYZ03.$UL.TEST17

    |             |   |      |    |                   |                  |       |
```

Field Descriptions

IPCDA	(IPCDATE) Date the unload or $DEFRAG processing occurred.
IPCTIME	Time the unload or $DEFRAG processing occurred.
IPCTYP	(IPCTYPE) IPC record type: D (Defrag volume record) or U (Unload record).
SC	Primary input source: BD (BILLDATA), MS (MSVOLGRP), VO (VOLUMES), OC (OSCATALOG), AC (ASM2CAT), SV (STGROUPS), DC (DATCLASS), MC (MGTCLASS), SC (STRCLASS), SG (STORGACD).
UNLRETPD	Retention period in days for the unloaded data set.
UNLVOL	Volume serial to which the data set was unloaded.
STRCLASS	SMS Storage Class name.
MGTCLASS	SMS Management Class name.
VOLUME	Volume serial number.
DSNAME	Name of the data set.

2.6.7.2 Example 2 - Searching the VMF and IPC in a Single Pass

This command searches the TLMS VMF and CA-ASM2 IPC for all data sets ending with the four character FILE.

```
$RSVP TLMSVMF ASM2CAT LEVEL(*) ENDING(FILE) -
PRINT(NEW (SOURCE IPCUTYPE TVOLSER TFILSEQ VOLUME ALLOC USED DSNAME))
```

```
$RS0A03 RSVP VERSION 1.0 $RS TLMSVMF ASM2CAT LEVEL(*) ENDING(FILE) PRINT(NEW (SOURCE IPCUTYPE TVOLSER TFILSEQ VOLU PAGE 1

SOURCE   IPCUTYP TVOLSER TFILSEQ VOLUME    ALLOC     USED DSNAME

TLMSVMF           ARC024      1               0        0 ASMBK.ANY.FILE
TLMSVMF           000105      1               0        0 ASM2.ANY.FILE
ASM2CAT  BACKUP              0  SJ0001      712       47 CN9000.AGS410.ARCH.SMONFILE
TLMSVMF           000115      1               0        0 HSM.ANY.FILE
TLMSVMF           ARC036      1               0        0 HSMBK.ANY.FILE
ASM2CAT  BACKUP              0  SJ0001      712      712 ABCDE01.ARCH.SMONFILE
TLMSVMF           TST033      1               0        0 ANONH01.COPYCAT.MVOL.FILE
TLMSVMF           TST032      1               0        0 ANONH01.COPYCAT.MVOL.FILE
TLMSVMF           TST031      1               0        0 ANONH01.COPYCAT.MVOL.FILE
TLMSVMF           TST029      1               0        0 ANONH01.COPYCAT.MVOL.FILE
TLMSVMF           TST028      1               0        0 ANONH01.COPYCAT.MVOL.FILE
TLMSVMF           TST027      1               0        0 ANONH01.COPYCAT.MVOL.FILE
TLMSVMF           TST015      1               0        0 ANONH01.COPYCAT.MVOL.FILE
TLMSVMF           TST014      1               0        0 ANONH01.COPYCAT.MVOL.FILE
TLMSVMF           TST013      1               0        0 ANONH01.COPYCAT.MVOL.FILE
TLMSVMF           TST030      1               0        0 ANONH01.COPYCAT.ONE.FILE
TLMSVMF           TST011      1               0        0 ANONH01.COPYCAT.ONE.FILE
TLMSVMF           000211      4               0        0 PROD1.BOBBLE.FILE
TLMSVMF           000211      1               0        0 PROD2.BOBBLE.FILE
TLMSVMF           000213      5               0        0 PROD2.SUM08.FILE
TLMSVMF           000212      5               0        0 PROD2.SUM08.FILE
TLMSVMF           000211      5               0        0 PROD2.SUM08.FILE
TLMSVMF           000213      7               0        0 PROD2.SUM14.FILE
TLMSVMF           000214      8               0        0 PROD2.SUM16.FILE
TLMSVMF           000213      8               0        0 PROD2.SUM16.FILE
TLMSVMF           000214      9               0        0 PROD2.SUM22.FILE
TLMSVMF           000214     10               0        0 PROD2.SUM23.FILE
TLMSVMF           000214     11               0        0 PROD2.SUM24.FILE
TLMSVMF           000214     12               0        0 PROD2.SUM38.FILE
TLMSVMF           000213      6               0        0 PROD3.SUM10.FILE
```

Field Descriptions

SOURCE Primary input source for the data returned.

IPCUTYP IPC unload record type.

TVOLSER Tape volume serial number.

TFILSEQ Tape file sequence number.

VOLUME Volume serial number.

ALLOC Data set allocation quantity.

USED Quantity of space used.

DSNAME Data set name.

2.6.7.3 Example 3 - Searching DASD Volumes and the IPC in a Single Pass

This command searches the DASD volumes beginning with the characters SJ and the CA-ASM2 IPC for data sets which begin with the high-level index of MURTO01 and contain the characters 410 anywhere in the data set name. The output is divided into sections based on the input source. Each input source is sorted separately.

```
$RSVP VOL(SJ) ASM2CAT LEVEL(WAZUP01) CONTAIN(410) CLUSTER NOMERGE  -
PRINT(NEW (SOURCE IPCUTYPE VOLUME ALLOC USED DSNAME)) NOTOTAL
```

```
$RS0A03 RSVP VERSION 1.0   $RS VOL(SJ) ASM2CAT LEVEL(MURTO01) CONTAIN(410) CLUSTER NOMERGE  PRINT(NEW (SOURCE IPCUTY  PAGE 1

SOURCE   IPCUTYP VOLUME    ALLOC     USED DSNAME

VOLUMES          SJ0001      475      475 WAZUP01.AGS410.ARCH.$ARQUEUE
VOLUMES          SJ0001       47       47 WAZUP01.AGS410.ARCH.$DUPPOOL
VOLUMES          SJ0002      427      427 WAZUP01.AGS410.ARCH.$RAQUEUE
VOLUMES          SJ0001      142       47 WAZUP01.AGS410.ARCH.ARCLOG
VOLUMES          SJ0001      712      617 WAZUP01.AGS410.ASM2JNL
VOLUMES          SJ0001      427      427 WAZUP01.AGS410.BKUP.$RAQUEUE
VOLUMES          SJ0001       47       47 WAZUP01.AGS410.BKUP.$TAPPOOL
VOLUMES          SJ0001      142      142 WAZUP01.AGS410.BKUP.ARCLOG
VOLUMES          SJ0001       47       47 WAZUP01.AGS410.BKUP.ARCPARMS
VOLUMES          SJ0001      237      237 WAZUP01.AGS410.BKUP.INCR
VOLUMES          SJ0005       47       47 WAZUP01.AGS410.BKUP.LODSYS
VOLUMES          SJ0005       47       47 WAZUP01.AGS410.BKUP.LODUSR
VOLUMES          SJ0005       47       47 WAZUP01.AGS410.BKUP.LOUSER
VOLUMES          SJ0005       47       47 WAZUP01.AGS410.BKUP.TAPELOG
VOLUMES          SJ0001      665       95 WAZUP01.AGS410.D082492
VOLUMES          SJ0001      665      380 WAZUP01.AGS410.JCL

   |              |        |        |     |     |    |

$RS0A03 RSVP VERSION 1.0   $RS VOL(SJ) ASM2CAT LEVEL(WAZUP01) CONTAIN(410) CLUSTER NOMERGE  PRINT(NEW (SOURCE IPCUTY  PAGE 2

SOURCE   IPCUTYP VOLUME    ALLOC     USED DSNAME

ASM2CAT  BACKUP  SJ0001     3561     3228 WAZUP01.AGS410.ASM
ASM2CAT  BACKUP  SJ0001      712      712 WAZUP01.AGS410.ASM2JNL.DATA

   |      |       |        |       |     |     |    |
```

Field Descriptions

SOURCE Primary input source for the data returned.

IPCUTYP IPC unload record type.

VOLUME Volume serial number.

ALLOC Data set allocation quantity.

USED Quantity of space used.

DSNAME Data set name.

2.6.8 Sort Examples

2.6.8.1 Example 1 - Using EXTSORT

This example shows the use of EXTSORT. If you do not specify BLIST containing the desired fields, the default field list, BILLLIST, is used for the transaction records ($RSTRAN) passed to a subsequent step. The NOPRINT keyword also causes the sort phase to be bypassed in the first step as if NOSORT were specified.

The sort program executed must correctly key in on the field that governs the order of the report's detail lines. That is, DSNAME is a 44-character long field beginning at column 56 of the transaction record produced by the SPACE field list. In the BILLLIST list, DSNAME would begin at column 23.

In the step that displays the records, INBILL must be specified with EXTSORT to show that the input was sorted in a previous step. INBLIST is also needed to define the field list of the input transaction file referenced by INBILL, otherwise, the default BILLLIST is used.

Sample JCL for this example is:

```
//JOBNAME JOB 'account no.'
//ASM2CMDU EXEC ASM2CMDU
//ASM2CMDU.$RSTRANS DD DSN=&&BLTRANS,DISP=(,PASS),
//             UNIT=SYSDA,SPACE=(TRK,(10,5))
//ASM2CMDU.SYSIN DD *
$RSVP VOL(SJO) NOPRINT BLIST(SPACE) TRKS
//*************************************************
//*      SAMPLE USE OF EXTERNAL SORT              *
//*************************************************
//SORT    EXEC PGM=SORT
//SORTLIB  DD  DSN=SYS1.SORTLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=(*)
//SYSOUT   DD  SYSOUT=(*)
//SYSIN    DD  *
   SORT  FIELDS=(56,44,CH,A)
//SORTWK01 DD  UNIT=SYSDA,SPACE=(TRK,(95),,CONTIG)
//Insert misc. SORTWKOx DDstmnts. here
//SORTIN   DD  DISP=(OLD,PASS),DSN=&&BLTRANS
//SORTOUT  DD  DISP=(NEW,PASS),DSN=&TEMP,
//             UNIT=SYSDA,SPACE=(CYL,(1,1))
//ASM2CMDU EXEC ASM2CMDU
//ASM2CMDU.BILL DD DISP=(OLD,PASS),DSN=&TEMP
//ASM2CMDU.SYSIN DD *
$RSVP INBILL(BILL) INBLIST(SPACE) -
PRINT(NEW (DSORG ALLOC USED EXT VOL DSN)) TRKS EXTSORT
```

2.6.8.2 Example 2 - Using Descending Sort

This example shows both a descending and ascending sort sequence for reporting. The field ALLOC is sort descending and DSNAME is ascending (by default). Observe that both a heading one and two are specified.

```
$RSVP VOL(TSO) PRINT(ADD (UNUSED PCT)) SORT(ALLOC(-) DSN) -
TRKS IF(ALLOC GT 5) AND1(PCT GT 90) AND2(DSORG EQ PO) -
HEADING1('1PO DATA SETS OVER 90% USED') -
HEADING2(' SORTED BY ALLOC(DESCENDING ORDER) AND DSNAME')
```

This command generates the following customized report:

```
PO DATA SETS OVER 90% USED                                                                          PAGE     1
SORTED BY ALLOC(DESCENDING ORDER) AND DSNAME                                                  13:09:45 MM/DD/YYYY
DSCOUNT DSORG LSTUS       ALLOC     USED   UNUSED PCT   EXT VOLUME DSNAME

         PO   04/23/00      614      609        5  99     1 TSO24A GALED02.CAI.CICSYYY.SOURCE
         PO   04/27/00      522      508       14  97     5 TSO22A SYSQA.TEST.CAILIB
         PO   02/23/00      451      451        0 100     2 TSO21A CAI.M21.CDS4200.F2
         PO   04/27/00      420      398       22  94     2 TSO22A SYSQA.TEST.SMPPTS
         PO   04/23/00      419      415        4  99     1 TSO24A CICS.MVS.V21.MACLIB
         PO   01/23/00      330      311       19  94     8 TSO21A CAI.CACT.ISPSLIB
         PO   02/27/00      330      316       14  95     8 TSO21A CAI.M21.CACT.ISPSLIB
         PO   06/13/00      330      306       24  92     1 TSO22A CAI.SYSQA.CACT.SO8905I3.ISPSLIB
         PO   03/22/00      330      302       28  91     1 TSO22A SYSQA.CACT.ISPSLIB
         PO   04/26/00      290      280       10  96     2 TSO24A A24.ROSCOE.PROD.MACROS
         PO   04/05/00      235      235        0 100     2 TSO21A CAI.M21.CDS4200.F1
         PO   04/23/00      225      221        4  98     1 TSO24A SD.ROGDA01.TAGMOD.LOAD
         PO   04/23/00      179      163       16  91     1 TSO24A SYSTEMS.ROS56.MACLIB
         PO   02/27/00      163      162        1  99     1 TSO21A CAI.M21.PPOPTION
         PO   04/23/00      161      161        0 100     6 TSO24A A24.ROSCOE.DIST.MACLIB.V560
         PO   04/23/00      154      154        0 100     5 TSO24A A24.ROSCOE.DIST.MACLIB.V570
         PO   04/23/00      132      121       11  91     1 TSO24A CICS.MVS.V21.LOADLIB
         PO   01/26/00      110      103        7  95     2 TSO21A CAI.V67.LOOK.OBJECT
         PO   04/23/00      105       99        6  94     1 TSO24A A24.ROSCOE.SUPPORT.DISTV56

         |       |          |        |        | |     |          |

         PO   04/27/00       25       23        2  92     3 TSO24A BARJO03.BAR.LOAD
         PO   02/23/00       25       25        0 100     1 TSO21A CAI.M21.CYV1400.F2
```

```
PO DATA SETS OVER 90% USED                                                                                  PAGE     2
SORTED BY ALLOC(DESCENDING ORDER) AND DSNAME                                                          13:09:45 MM/DD/YYYY
DSCOUNT DSORG LSTUS       ALLOC     USED   UNUSED PCT   EXT VOLUME DSNAME

         PO   01/23/00       25       25        0 100     1 TSO21A CAI.M21.CZ26000.F2
         PO   02/21/00       23       23        0 100     1 TSO21A CAI.M21.CJR6008.F1
         PO   01/23/00       22       22        0 100     1 TSO21A CAI.M21.CZ16000.F2
         PO   02/27/00       21       21        0 100     1 TSO21A CAI.EARL.CEO50LLD
         PO   02/27/00       21       20        1  95     4 TSO21A CAI.INSTALL
         PO   02/21/00       21       21        0 100     1 TSO21A CAI.M21.CJW2300.F2
         PO   02/21/00       20       20        0 100     1 TSO21A CAI.M21.CJR6008.F2
         PO   02/21/00       19       19        0 100     1 TSO21A CAI.M21.CJ16002.F1
         PO   04/27/00       19       18        1  94     1 TSO24A DATACOM.DAF.YU.DV210.XEQLIB
         PO   04/27/00       19       18        1  94     1 TSO03A MOSEF01.CBASESYS.CBASE.V1R1M1.TEMPLATE
         PO   02/21/00       17       17        0 100     1 TSO21A CAI.M21.CJR6000.F2
         PO   02/21/00       17       17        0 100     1 TSO21A CAI.M21.CJR6001.F1
         PO   02/21/00       14       14        0 100     1 TSO21A CAI.M21.CJR6002.F2
         PO   02/21/00       14       14        0 100     1 TSO21A CAI.M21.CJR6005.F2
         PO   02/21/00       14       14        0 100     1 TSO21A CAI.M21.CJR6006.F2
         PO   02/21/00       14       14        0 100     1 TSO21A CAI.M21.CJR6007.F1
         PO   02/21/00       14       14        0 100     1 TSO21A CAI.M21.CJR6007.F2

          |       |          |        |         | |      |           |

         PO   04/23/00       12       12        0 100     1 TSO24A DICAN01.TEMP.LOAD
         PO   02/27/00       10       10        0 100     2 TSO21A CAI.M21.CAICLIB
         PO   04/27/00        8        8        0 100     1 TSO24A DATACOM.DAF.YU.DC240.XEQLIB
         PO   04/23/00        7        7        0 100     1 TSO24A SD.ROGDA01.DYNLIB.LOAD
    100                    8177     7906      271       253
```

2.6.9 Output Format Examples

2.6.9.1 Example 1 - Using the PRINT Keyword to Change Formats

Example 1.1 Using DEL

This command deletes the DSCOUNT and USED fields from the default print list (SPACE). To delete only one field, the inner pair of parentheses is not required; that is, PRINT(DEL DSCOUNT) is correct. The VOL keyword limits the volumes to be searched, and the IF and AND keywords further limit data set selection to data sets with allocation quantities greater than 300 and extents greater than one.

```
$RSVP VOL(USER) IF(ALLOC GT 300) AND1(EXT GT 1) LIST(SPACE) -
PRINT(DEL (DSCOUNT USED))
```

This generates the following modified-standard report:

```
$RS0A03 RSVP VERSION 1.0   $RSVP VOL(USER)          PAGE  1
DSORG LSTUS    ALLOC UNUSED EXT VOLUME DSNAME

 PS   06/09/00  1087    324   3 USER02 $CAI00.$AI.LIST
 PS   06/09/00   477     19   2 USER02 $CAI00.$CRTEXPR.LIST
 PS   06/09/00   610      0   2 USER02 $CAI00.$DA.LIST
 PS   06/09/00   400      0   2 USER02 $CAI00.$NEWPARS.LIST
 PO   06/10/00  1278    439   2 USER02 $CAI00.AJ.CNTL

 |    |           |      |   | |      |

 PS   05/19/00   305     95   2 USER01 $CAI98.L703.VTOC.DATA
 PS   05/19/00   305     76   2 USER01 $CAI98.T703.VTOC.CARDS
 PO   05/19/00   420    191   2 USER02 $CAI99.MISC.OBJ
 PO   05/19/00   801    343   3 USER02 $CAI99.R250703.OBJ
 PO   05/19/00   400    133   2 USER02 $CAI99.R250708.OBJ
 PO   05/19/00   477    324   2 USER02 $CAI99.SOURCE.OBJ
              48223  13668 212        TOTAL
```

Example 1.2 Using ADD

This command adds fields to the print list (SPACE). The first field indicates the field after which additional fields are to be added. In this example, LMDATE is added immediately after LSTUS.

```
$RSVP VOL(USER) IF(ALLOC GT 300) AND1(EXT GT 1) LIST(SPACE) -
PRINT(ADD (LSTUS LMDATE))
```

This generates the following modified-standard report:

```
$RS0A03 RSVP VERSION 1.0     $RSVP VOL(USER)       PAGE 1
DSCOUNT DSORG LSTUS    LMDATE   ALLOC  USED UNUSED EXT ...

         PS   06/09/99 06/09/00  1087   763   324    3 ...
         PS   06/09/99 06/09/00   477   458    19    2 ...
         PS   06/09/99 06/09/00   610   610     0    2 ...
         PS   06/09/99 06/09/00   400   400     0    2 ...
         PO   06/10/99 06/10/00  1278   839   439    2 ...
         PO   06/06/99 05/31/00  2860  2345   515    5 ...

         |    |       |           |     |     |    | |

         PS   05/19/99 12/04/00   305   210    95    2 ...
         PS   05/19/99 12/04/00   305   229    76    2 ...
         PO   05/19/99 05/01/00   420   229   194    2 ...
         PO   05/19/99 04/24/00   801   458   343    3 ...
         PO   05/19/99 04/22/00   400   267   133    2 ...
         PO   05/19/99 03/28/00   477   153   324    2 ...
     45                         48223 34555 13668  212 ...
```

Example 1.3 Using REP

This command replaces one field in the default list (SPACE) with multiple fields that are not normally present. The field DSCOUNT is replaced with RECFM, BLKSZ, and LRECL.

```
$RSVP VOL(USER) IF(ALLOC GT 300) AND1(EXT GT 1) -
PRINT(REP (DSCOUNT RECFM BLKSZ LRECL))
```

This generates the following modified-standard report:

```
$RS0A03 RSVP VERSION 1.0     $RSVP VOL(USER) IF(ALLOC PAGE 1
RECFM BLKSZ LRECL DSORG LSTUS    ALLOC  USED UNUSED EXT ...

FBA     121   121  PS   06/09/90  1087   763    324   3 ...
FBA     121   121  PS   06/09/90   477   458     19   2 ...
FBA     121   121  PS   06/09/90   610   610      0   2 ...
FA      121   121  PS   06/09/90   400   400      0   2 ...
FB     3120    80  PO   06/10/90  1278   839    439   2 ...
FB     3600    80  PO   06/06/90  2860  2345    515   5 ...

|        |     | |    |           |     |      |   | |

FB     1400   140  PS   05/19/90   305   210     95   2 ...
FB     6160    80  PS   05/19/98   305   229     76   2 ...
FB       80    80  PO   05/19/90   420   229    194   2 ...
FB       80    80  PO   05/19/90   801   458    343   3 ...
FB       80    80  PO   05/19/90   400   267    133   2 ...
FB       80    80  PO   05/19/90   477   153    324   2 ...
                              48223 34555  13668 212 ...
```

Example 1.4 Using NEW

This command replaces the default print list with a completely new list. The new list prints fields DSNAME, LSTUS, LMDATE, ALLOC, PCT, and EXT.

```
$RSVP VOL(USER) IF(ALLOC GT 300) AND1(EXT GT 1) -
PRINT(NEW (DSNAME LSTUS LMDATE ALLOC PCT EXT))
```

This generates the following customized report:

```
$RS0A03 RSVP VERSION 1.0              $RSVP VOL(USER)     PAGE  1
DSNAME                             LSTUS    LMDATE   ALLOC PCT EXT

$CAI00.$AI.LIST                    06/09/00 06/09/00  1087  70   3
$CAI80.$CRTEXPR.LIST               06/09/00 06/09/00   477  96   2
$CAI00.$DA.LIST                    06/09/00 06/09/00   610 100   2
$CAI00.$NEWPARS.LIST               06/09/00 06/09/00   400 100   2
$CAI00.AJ.CNTL                     06/10/00 06/10/00  1278  65   2
$CAI00.CDG.ASM                     06/06/00 05/31/00  2860  81   5
$CAI04.MIGRATE.SR022880            05/19/00            667 100   3

      |                               |      |       |   |    |

$CAI98.L703.VTOC.DATA              05/19/00 12/04/98   305  68   2
$CAI98.T703.VTOC.CARDS             05/19/00 12/04/98   305  75   2
$CAI99.MISC.OBJ                    05/19/00 05/01/00   420  54   2
$CAI99.R250703.OBJ                 05/19/00 04/24/00   801  57   3
$CAI99.R250708.OBJ                 05/19/00 04/22/00   400  66   2
$CAI99.SOURCE.OBJ                  05/19/00 03/28/00   477  32   2
TOTAL                                                 48223     212
```

2.6.9.2 Example 2 - Collecting Space Management Information

Example 2.1 DASD Use by User ID (High-level Index)

This example uses the keyword INDEXTOTALS to determine the amount of DASD space being used by each user. This keyword specifies the number of index levels on which to subtotal. The default of SORT(DSNAME) is assumed when the SORT keyword is not specfied.

```
$RSVP VOL(*) INDEXTOTALS(1) TRK -
NOPRINT(NEW (DSCOUNT ALLOC USED UNUSED EXT DSNAME))
```

This command generates a summary of data set usage by high-level indexes since INDEXTOTALS specifies one level. Since 1 is the default for INDEXTOTALS, you could omit the (1) without changing the meaning of the command. The keyword NOPRINT eliminates detail lines since you are only interested in total space usage. But you can still enter fields to control subtotal and total print lines.

This generates the following report:

```
$RS0A03 RSVP VERSION 1.0 $RSVP VOL(*) INDEX(1) PAGE  1
DSCOUNT    ALLOC    USED    UNUSED    EXT    DSNAME

     73     1813     1313       500    148    $CAI00.
     33     1045      683       362     60    $CAI01.
     77      758      383       375    115    $CAI02.
     55     1081      572       509    125    $CAI03.
     25      513      426        87     38    $CAI04.
     23      845      458       387     26    $CAI05.
      2       32       23         9      4    $CAI98.
     88     4687     3687      1000    140    $CAI99.
    376    10774     7545      3229    656    TOTAL
```

Example 2.2 Online Volume Free Space

To ensure there is sufficient space for a particularly large job to run, it is convenient to know the amount of free space on volumes currently online. In this example, the LIST keyword and the VOLUME list show the volume free space and largest extents.

```
$RSVP VOL(*) TRK LIST(VOLUME)
```

This command generates the following standard report:

```
$RS0A03 RSVP VERSION 1.0            $RSVP VOL(*)          PAGE  1
VOLUME   CUU   UNIT   DEVTYPE   VOLATTR   VOLSPC   VOLEXT  VOL1XSP

TSG001   130   3390   3010200E  PRIVATE     1640     0003      572
SYS100   132   3390   3010200E  PRIVATE    41475     0008    34324
SYS500   239   3390   3010200E  PRIVATE    14759     0020     2860
SYS300   331   3390   3010200E  PRIVATE   131481     0009   102286
USER01   332   3390   3010200E  PRIVATE    13119     0150     1144
SYS600   336   3390   3010200E  PRIVATE    85219     0001    85219
SYS200   538   3390   3010200E  PRIVATE     7075     0017     1030
SYS400   539   3390   3010200E  PRIVATE        0     0000        0
USER02   53A   3390   3010200E  PRIVATE    11213     0089     1888
TSG002   53E   3390   3010200E  PRIVATE   108712     0038    50342
DEV002   53F   3390   3010200E  PRIVATE    16571     0029    10297
```

Example 2.3 Data Sets with Multiple Extents

In this example, the IF keyword is used to find data sets that are going into multiple extents since they cause fragmentation. This keyword selects data sets with the field named EXT greater than one.

```
$RSVP VOL(USER) IF(EXT GT 1) PRINT(DEL DSCOUNT)
```

This generates the following modified-standard report:

```
$RS0A03 RSVP VERSION 1.0     $RSVP VOL(USER)      PAGE  1
DSORG LSTUS   ALLOC  USED UNUSED EXT VOLUME DSNAME

 PS  06/09/00  1087   763    324   3 USER02 $CAI00.$AI.LIST
 PS  06/09/00   477   458     19   2 USER02 $CAI00.$CR.LIST
 PS  06/09/00   610   610      0   2 USER02 $CAI00.$DA.LIST
 PS  06/09/00   400   400      0   2 USER02 $CAI00.$PR.LIST
 PO  06/10/00  1278   839    439   2 USER02 $CAI00.AJ.CNTL
 PO  06/06/00  2860  2345    515   5 USER02 $CAI00.CDG.ASM

 |      |        |     |      |    |   |          |

 PS  05/19/00   305   210     95   2 USER01 $CAI98.VT.DATA
 PS  05/19/00   305   229     76   2 USER01 $CAI98.VT.CARDS
 PO  05/19/00   420   229    194   2 USER02 $CAI99.MISC.OBJ
 PO  05/19/00   801   458    343   3 USER02 $CAI99.R253.OBJ
 PO  05/19/00   400   267    133   2 USER02 $CAI99.R258.OBJ
 PO  05/19/00   477   153    324   2 USER02 $CAI99.S.OBJ
             50967 36573  14394 298        TOTAL
```

Example 2.4 Locating a Bad Track

If a DASD device shows a bad track (from LOGREC or a full pack backup such as $DEFRAG, FDR or IEHDASDR), it is important to know which data set contains the bad track. RSVP can provide this information by using the IF keyword to look for that address in CCHH format. The field CCHH is in hex format.

```
$RSVP VOL(USER02) IF(CCHH EQ 1D001B) -
PRINT(NEW (DSNAME CCHH EXTSEQ ALLOC))
```

This generates the following modified-standard report:

```
$RS0A03 RSVP VERSION 1.0   $RSVP VOL(USER02) PAGE  1
DSNAME               CCHH             EXTSEQ ALLOC

$DPC04.GORILLA.TEXT  001D0000 001D001D    0    572
TOTAL                                          572
```

Example 2.5 Volumes Running Out of Free Space

This example uses the same VOLUME list as Example 2.2, but uses the IF keyword to select volumes short on free space.

```
$RSVP VOL(*) IF(VOLSPC LT 100) TRK LIST(VOLUME)
```

This command generates the following standard report:

```
$RS0A03 RSVP VERSION 1.0     $RSVP VOL(*)      PAGE 1
VOLUME CUU UNIT DEVTYPE  VOLATTR VOLSPC VOLEXT VOL1XSP

TSG001 130 3390 3010200E PRIVATE     86  0003       30
SYS400 539 3390 3010200E PRIVATE      0  0000        0
```

Example 2.6 Find $DEFRAG Candidate Volumes

To avoid fragmentation on volumes, the CA-ASM2 $DEFRAG component reorganizes volumes to maintain a large free space extent. RSVP can identify volumes that should be processed with $DEFRAG. You can use the IF and AND keywords to select volumes with a large number of free-space extents (VOLEXT) and a reasonable number of free tracks (VOLSPC).

```
$RSVP VOL(*) TRK LIST(VOLUME)     -
IF(VOLEXT GT 25) AND1(VOLSPC GT 500)
```

This command generates the following standard report:

```
$RS0A03 RSVP VERSION 1.0   $RSVP VOL(*)       PAGE  1
VOLUME CUU UNIT DEVTYPE  VOLATTR VOLSPC VOLEXT VOL1XSP

TEST01 133 3390 3010200E STORAGE   9341    29    5670
WORK02 238 3390 3010200E STORAGE   4651    40    1320
USER01 332 3390 3010200E PRIVATE    871   158      60
WORK01 333 3390 3010200E STORAGE   9435    38    4380
SFI001 334 3390 3010200E PRIVATE   2173    47    1290
DEV001 337 3390 3010200E PRIVATE   3574    39    1200
TSG002 53E 3390 3010200E PRIVATE   5431    38    2640
DEV002 53F 3390 3010200E PRIVATE   2730    32    1835
```

Note: VOLSPC reports free space in a different format than the Realtime Space Monitor.

2.6.10 VSAM Reporting Examples

2.6.10.1 Example 1 - Using VSAMCAT to Reference a DDNAME

The following JCL and command illustrate the use of the VSAMCAT keyword to refer to a DDNAME in the JCL. This command also specifies DATA to print entries at the component level.

```
//VSAM     EXEC ASM2CMDU
//SYSUDUMP DD SYSOUT=A
//CATALOG  DD DSN=USERCAT.VUSER01,DISP=SHR
//SYSIN    DD *
$RSVP VOL(USER) TRK IF(DSORG EQ VS)  -
PRINT(REP (DSCOUNT VS))              -
VSAMCAT(CATALOG) DATA
```

This generates the next report. Since entities on the volumes being processed are related to a catalog other than the one specified, the error messages shown occur.

```
$RS0A03 RSVP VERSION 1.0     $RSVP VOL(USER) TRK PAGE  1
VS DSORG LSTUS    ALLOC USED UNUSED EXT VOLUME DSNAME

D   VS   06/17/00   180  151     29   1 USER02 V$CAI.DATA
D   VS   06/17/00   180  180      0   1 USER01 V$CAI.DATA
I   VS   06/17/00     1    1      0   1 USER01 V$CAI.INDEX
D   VS                60    1     59   1 USER02 VSAMDSET...
D   VS   06/05/00    60   46     14   1 USER02 VSAMDSET...
                    481  379    102   5        TOTAL




 $VS4802 VSAM CATALOG MANAGEMENT RETURN CODE -
         IN MODULE IGG0CLAH: 0004-010 FOR OBJECT: USER01
 $RS8K02 VSAM ERROR ON  USER01 -
         Z9999994.VSAMDSPC.T8F61E7F.TE443BE0
 $VS4802 VSAM CATALOG MANAGEMENT RETURN CODE -
         IN MODULE IGG0CLAH: 0004-010 FOR OBJECT: USER01
 $RS8K02 VSAM ERROR ON  USER01 -
         VSAMDSET.T3D74C10.DFD80146.T9037BAA.T3D74C10
 $VS4802 VSAM CATALOG MANAGEMENT RETURN CODE -
         IN MODULE IGG0CLAH: 0004-010 FOR OBJECT: USER01
 $RS8K02 VSAM ERROR ON  USER01 -
         VSAMDSET.T3D754E0.DFD80146.T9037BAA.T3D754E0
```

2.6.10.2 Example 2 - Using VSAMCAT to Reference a VSAM Catalog

In this example, VSAMCAT refers to the actual VSAM catalog to be searched. The elements are listed at the DATA level. But this time the extents are mapped by the LIST(MAP).

```
$RSVP VOL(PAG002) TRK IF(DSORG EQ VS) LIST(MAP) -
VSAMCAT(CATALOG.VPAG001) DATA -
PRINT(REP (DSCOUNT VS))
```

This command generates the following modified-standard report:

```
$RS0A03 RSVP VERSION 1.0     $RSVP VOL(PAG002) PAGE  1
VS DSNAME             ABSTR ABLEN ALLOC CCHH EXTSEQ EXT

D  PAGE.COMMON        10590   150   150 ...      1   1
D  PAGE.DUPLEX        13620   360   360 ...      1   1
D  PAGE.PLPA             30   450   450 ...      1   1
I  VSYS4.ALTINFO      11362     2   122 ...      1   2
D  VSYS4.ALTINFO      12870   120   122 ...      1   2
I  VSYS4.ALTLIDS      12148     2   122 ...      1   2
D  VSYS4.ALTLIDS       6150   120   122 ...      1   2
I  VSYS4.ALTRULES      6571     2   122 ...      1   2
D  VSYS4.ALTRULES     12750   120   122 ...      1   2
I  VSYS4.IS.VSAM       6540     4  5866 ...      3   8
I  VSYS4.IS.VSAM      12566     4  5866 ...      2   8
I  VSYS4.IS.VSAM       5670    38  5866 ...      1   8
D  VSYS4.IS.VSAM      15780   450  5866 ...      5   8
D  VSYS4.IS.VSAM      15330   450  5866 ...      4   8
D  VSYS4.IS.VSAM      14880   450  5866 ...      3   8
D  VSYS4.IS.VSAM       7590   180  5866 ...      2   8
D  VSYS4.IS.VSAM        480  4290  5866 ...      1   8
   TOTAL                          48620             79
```

Chapter 3. Preprocessing

Preprocessing is defined as any preliminary processing step that sets the stage for actual processing to be performed. Using RSVP for preprocessing, you define installation storage management criteria and subsequent management actions. Parameter keywords control entries to (1) the transaction file, which may be used as input to another program or another operation, or (2) CA-ASM2 queues. Once you define the criteria, data sets that meet the criteria are automatically selected. Selected data sets are often those that violate installation standards in placement, size, excess space or period of activity. Your response depends on the situation and your installation standards. Some actions might be to migrate the data to another volume, archive or delete it, compress it and release excess space, or just report on the exceptions. RSVP can assist you in performing any or all of these processes and make them totally automatic.

Some typical applications of preprocessing include:

- Generating space management commands to compress selected partitioned data sets.
- Generating commands to release unused space for overallocated data sets.
- Generating IEHPROGM SCRATCH commands to scratch uncataloged data sets.

Boolean operators can be used to test fields in the Format-1 DSCB.

Transaction files are created in any format to meet selection criteria. It takes no direct action on selected data sets, but rather produces commands for further processing. You can review or modify these commands prior to actual execution.

A report can be issued at the same time transaction commands are generated for selected data sets, thus providing a diagnostic report of the actions to be performed on all data sets.

Several default transaction formats (see BLIST on page 3-5) satisfy most installation requirements. You can easily customize additional formats to meet your particular requirements.

DASD preprocessing has the capability to be partially or completely driven by SMS information. With SMS, data management rules are stored and utilized for determining data management activities. Figure 4 illustrates the relationship between SMS and RSVP.

RSVP can interrogate the rules in the SMS database, analyze the appropriate volumes and data sets to determine the required data management operations, and pass either command or queued requests to the appropriate data management function.

The possible uses of a preprocessor are limited only by your imagination. The examples in this chapter give you an idea of some possibilities, but your installation can ultimately set up its own uses. Most or all of your normal maintenance can be scheduled as a series of $RSVP preprocessing commands, greatly reducing the time required to keep volumes clean.

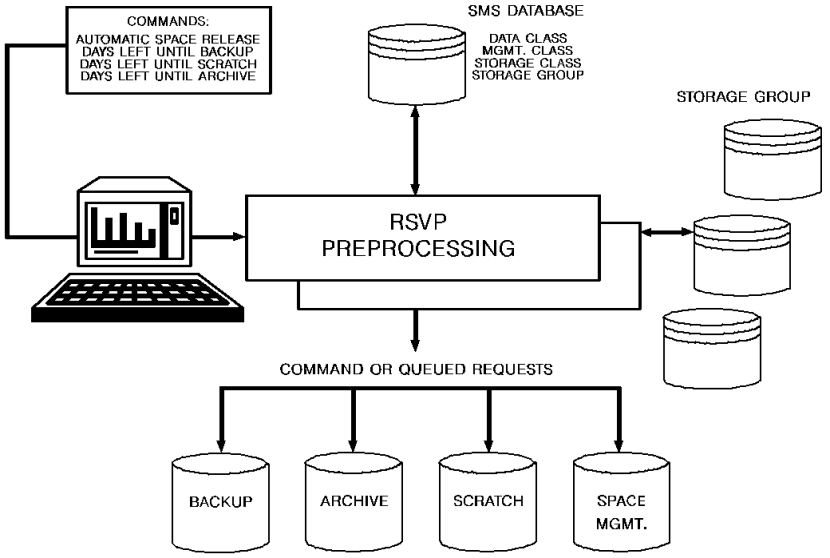


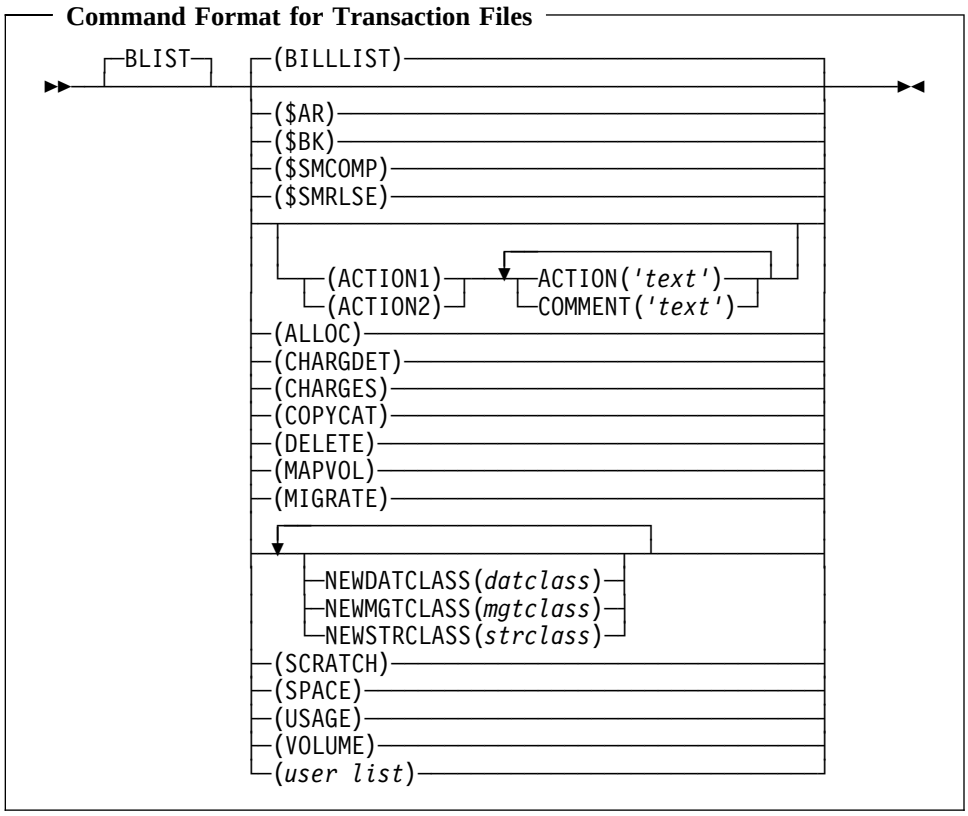
Figure 4 - RSVP and SMS

3.1 Transaction Files

RSVP preprocessing writes a record to the output transaction file, $RSTRANS, for each data set selected. $RSTRANS is then used as input to another program, usually in the next step. A $RSTRANS DD statement is required to use RSVP for preprocessing. You can execute the ASM2CMDU procedure and the IBM utility IEHPROGM for processing commands.

In transaction files, character fields are left aligned and padded with blanks, and integers are right aligned and prefixed with blanks; if not, fields are the correct length.

3.1.1 Command Format for Transaction Files



Keyword Descriptions for Transaction Files:

For examples showing the use of these keywords, see Transaction File Examples on page 3-12.

The following literals can be placed into a transaction record:

Literal	Description
LPAREN	Defines an open left parentheses (literal field): (
RPAREN	Defines a closed right parentheses (literal field):)
QUOTE	Defines a single quote (literal field): '
L$ARDSN	Defines a literal string: $AR DA(dsname
L$BKDSN	Defines a literal string: $BK DA(dsname
L$SMDSN	Defines a literal string: $SM DA(dsname
L$RLSE	Defines a literal string: RLSE
L$RPVOL	Defines a literal string:) VOL(

See the BLIST keyword ($AR, $BK, and $SMRLSE) for examples using these literals.

The following keywords control what information RSVP writes to the transaction file.

Keyword	Description
ACTION	Defines a character string to be inserted with other fields as part of the records written to the transaction file. The operand must be enclosed in single quotes. For example: ACTION('character string')
COMMENT	Defines an additional string that you may include as part of the output records written to the transaction file. The operand must be enclosed in single quotes. For example: COM('additional character string') **Note:** The single quotes (') in ACTION or COMMENT must be balanced. Since single quotes are needed on the generated command, you must use two single quotes (") to get one single quote passed on to the output command. For example: `BLIST(ACTION1) ACTION('$AR ''') COMMENT('''')` generates $AR 'dsname'
NEWDATCLASS	Specifies a 1- to 8-character SMS Data Class name.
NEWMGTCLASS	Specifies a 1- to 8-character SMS Management Class name.
NEWSTRCLASS	Specifies a 1- to 8-character SMS Storage Class name.

BLIST Specifies the name of an existing, stored field list used to build output transactions. In addition to the field lists defined by the LIST keyword, additional field lists useful for preprocessing are:

List	Fields for Transaction Files
ACTION1	ACTION, DSNAME, COMMENT
ACTION2	ACTION, VOLUME, COMMENT, DSNAME
MIGRATE	DSNAME, VOLUME, NEWSTRCL, NEWMGTCL, NEWDATCL
$AR	L$ARDSN,QUOTE,DSNAME,QUOTE,L$RPVOL, VOLUME,RPAREN
$BK	L$BKDSN,QUOTE,DSNAME,QUOTE,L$RPVOL, VOLUME,RPAREN For example, $BK would generate the following: `$BK('DATA.SET.NAME') VOL(SYS002)`
$SMRLSE	L$SMDSN,QUOTE,DSNAME,QUOTE,L$RPVOL, VOLUME,RPAREN,L$RLSE
$SMCOMP	'$SM DSNAME('',DSNAME,'') VOL(',VOLUME,) COMPRESS RLSE PCTSAVE(10)'
SCRATCH	'SCRATCH VOL=',UNIT,'=',VOLUME,',',@72,'X' @16,'DSNAME=',DSNAME

DELETE	'SCRATCH VOL=',UNIT,'=',VOLUME,',',@72,'X' @16,'DSNAME=',DSNAME,'UNCATLG DSNAME=',DSNAME
	For example, DELETE would generate the following:
	`SCRATCH VOL=unit=volser,` X
	`DSNAME=dsname`
	`UNCATLG DSNAME=dsname`
	Column 72 of the first record contains the literal X and the second record begins in column 16 with DSNAME=.

COPYCAT	TVOLSER

3.1.2 Generating Transaction File Commands

The command BLIST($AR) causes the transaction file to contain $AR commands in which the spaces in the 44-byte DSNAME are suppressed, whereas:

```
BLIST(ACTION2) ACTION(' SCRATCH VOL=3390=') -
COMMENT(',DSNAME=')
```

generates this IEHPROGM SCRATCH command:

```
SCRATCH VOL=3390=volume,DSNAME=dsname
```

The command SUBCOMMAND($AR/$BK) can also be used to specify $AR or $BK commands (see 3-7).

Another method of generating SCRATCH commands is with BLIST(SCRATCH), which generates two-line scratch commands in the form:

```
SCRATCH VOL=unit=volser,                                X
              DSNAME=dsname
```

A similar keyword is BLIST(DELETE), which creates the SCRATCH commands and also generates UNCATLG commands. The format of the output command is:

```
SCRATCH VOL=unit=volser,                                X
               DSNAME=dsname
UNCATLG DSNAME=dsname
```

Although the BLISTs just listed cover most possibilities, it is fairly simple to add new lists and predefine other commands to avoid having to enter long commands all the time (see Output Format Phase on page 2-20 in the "Reporting" chapter for details). See User CSECT on page 5-9 chapter for information on tailoring RSVP.

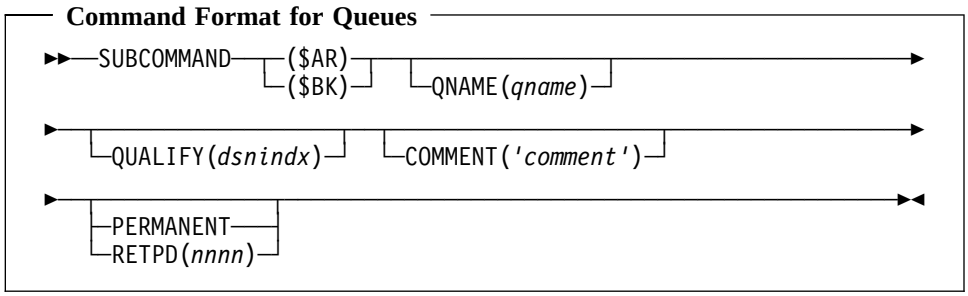
The $RSTRANS file is record length 132, but IBM utilities accepts only 80-byte card images for input. To create an 80-byte record, specify USERCSECT($RSUSER8). This causes a different user CSECT to be loaded that specifies a record length of 80 and a block size of 6160 for $RSTRANS. See the User CSECT on page 5-9 chapter for more information on the user CSECT.

3.2 Archive and Backup Queues

Besides generating commands in transaction files, RSVP can place data sets directly into the archive or backup queue.

When you use this feature, a report identifying the data sets satisfying the selection criteria is issued. The data set names and other information identifying the data sets are placed in either the archive or backup queue, as appropriate. If desired, you can issue a $QM command to verify the existence of the data set in the queue (see the *CA-ASM2 System Reference Guide* for $QM command syntax).

3.2.1 Command Format for Queues



Keyword Descriptions for Queues:

An example showing the use of these keywords is shown on 3-21.

SUBCOMMAND Specifies whether $AR or $BK commands are being created. These commands place the identified data sets in the archive or backup queue, as appropriate. When the queues are processed, the data sets are actually archived or backed up. For example, SUBCOMMAND($AR) or SUBCOMMAND($BK).

QNAME(qname) Places the requests in a logical queue of requests with the qname specified. Data sets archived (or backed up) with the same qname are unloaded onto the same tape.

QUALIFY(dsnindx) Renames the specified data sets by adding the dsnindx to the existing name when archived (or backed up). Restoration must be under the new name. The dsnindx is a 1- to 8-character string.

COMMENT(comment)
Specifies up to 30 characters as comments that are preserved in the IPC. The operand must be enclosed in single quotes.

PERMANENT	Places the data set on a permanent archive tape.
RETPD(nnnn)	Specifies the number of days (up to your installation-set value) to retain the data on tape. A value of 9999 sets RETPD to your installation-defined maximum.

3.3 Controlling Unregistered Data Sets

RSVP allows you to register data set names and the volume on which they must reside. Registration is simply a way of assigning data sets to accounts for billing purposes. This process is described in Billing on page 4-2 in the "DASD Billing" chapter. You can eliminate any data set that fails to meet the registration requirement.

The procedure to scratch unregistered data sets involves these three steps:

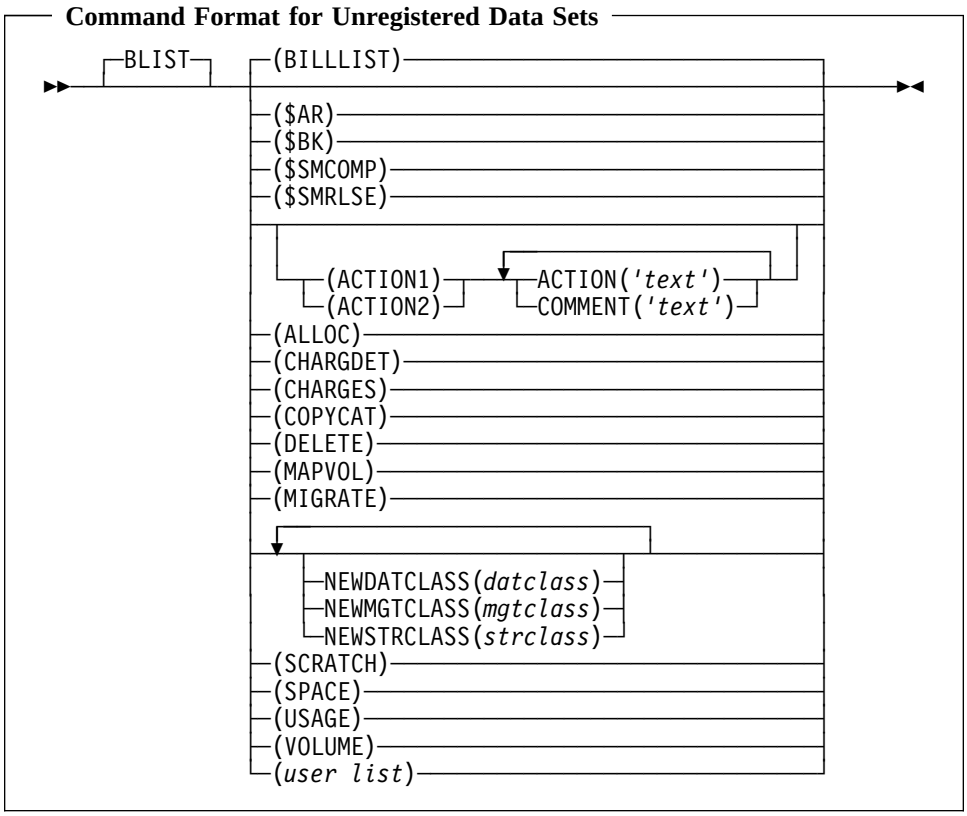
1. Data sets are matched to a registration file.
2. Data sets are selected that do not match.
3. A utility performs the actual maintenance.

The command might look like this:

```
$RSVP INBILL(INTRAN) INBLIST(BILLLIST) BLIST($AR) -
IF(ACCOUNT EQ UNKNOWN) NOSORT
```

This command indicates that the input from the first step is in the data set referenced by DDNAME INTRAN. It was created using BILLLIST as the field list. The command output (to $RSTRANS) is $AR commands. And, any input transaction record that has an account number of UNKNOWN is selected for archival. You could use this procedure to scratch the unregistered data sets instead of archiving them.

3.3.1 Command Format for Unregistered Data Sets



Keyword Descriptions for Unregistered Data Sets:

For examples using these keywords, see Unregistered Data Sets Examples on page 3-22.

BLIST Specifies the name of an existing, stored field list used to build output transactions.

List	Fields
$AR	'$AR,DSNAME,('',DSNAME,'') VOL(',VOLUME,RPAREN
$BK	'$BK,DSNAME,('',DSNAME,'') VOL(',VOLUME,RPAREN
$SMCOMP	'$SM DSNAME('',DSNAME,'') VOL(',VOLUME,) COMPRESS RLSE PCTSAVE(10)'
$SMRLSE	'$SM DSNAME('',DSNAME,'') VOL(',VOLUME,') COMPRESS RLSE PCTSAVE(10)'
ACTION1	ACTION, DSNAME, COMMENT
ACTION2	ACTION, VOLUME, COMMENT, DSNAME

ALLOC DSCOUNT, ALLOC, USED, UNUSED, PCT, EXT, SECQ, SECT, DSNAME, VOLUME, DSORG, RECFM, BLKSZ, LRECL, CREDT, LSTUS

BILLLIST DATE, UNIT, VOLUME, DSNAME, ACCOUNT, KBALLOC, KBDAYS, CHCUR

CHARGDET ACCOUNT, DSCOUNT, KBALLOC, KBDAYS, CHCUR, CHMTD, CHYTD, UNIT, VOLUME, DSNAME

CHARGES ACCOUNT, DSCOUNT, ALLOC, KBDAYS, CHCUR, CHMTD, CHYTD

HISTORY DATE, CHYTD, CHMTD, CHMONTHS, ACCOUNT

SCRATCH 'SCRATCH VOL=',UNIT,'=',VOLUME,',',@72,'X' @16,'DSNAME=',DSNAME

SPACE DSCOUNT, DSORG, LSTUS, ALLOC, USED, UNUSED, EXT, VOLUME, DSNAME

user list A user-supplied list of report fields.

INBILL Specifies the DDNAME of an existing transaction file from a previous $RSVP command (that is, this keyword is used to reference the output from the first step).

Note: INBILL cannot refer to $RSTRANS since you are writing to $RSTRANS.

INBLIST Specifies the field list used to create the existing transaction file referenced in INBILL. If you enter keyword BLIST, INBLIST defaults to the same list. If neither BLIST nor INBLIST are specified, the default is BILLLIST. The default can be changed in the user CSECT (see 5-9 in the "Tailoring" chapter).

3.4 Examples

The following JCL and report examples are presented in this section:

- Transaction File
- Archive and backup queues
- Unregistered data sets

3.4.1 Transaction File Examples

3.4.1.1 Example 1 - Generating $SM Commands to Release Space

Example 1.1 Building a Command with ACTION1 BLIST

You can recover overallocated space in sequential and partitioned data sets by issuing $SM commands to release the excess space. RSVP can assist by generating the $SM commands. In this example, a space report is generated and a transaction file of $SM commands for all sequential data sets on volumes whose VOLSERs begin with USER having more than five unused tracks. Then another step reads the transaction file and executes the generated commands.

```
//GENTSO    EXEC ASM2CMDU
//$RSOUT    DD SYSOUT=*
//$RSTRANS  DD DSN=&&TSOSM,DISP=(,PASS),
//             UNIT=SYSDA,SPACE=(TRK,(10,5))
//SYSIN     DD *
$RSVP VOL(USER) IF(DSORG EQ PS) AND1(UNUSED GT 5) TRK   -
PRINT(DEL DSCOUNT) TOTALS                               -
ACTION('$SM DSNAME(''') COM(''') RLSE ') BLIST(ACTION1)
/*
//TSOSM     EXEC ASM2CMDU
//SYSIN     DD DSN=&&TSOSM,DISP=(OLD,DELETE)
```

The report generated lists the data sets to have space released.

```
$RS0A03 RSVP VERSION 1.0   $RSVP VOL(USER)           PAGE 1
DSORG LSTUS    ALLOC USED UNUSED EXT VOLUME DSNAME

 PS   06/09/00    57   40     17   3 USER02 $CAI00.$AI.LIST
 PS   05/19/00    60   42     18   2 USER02 $CAI00.D.DATA
 PS   05/27/00    16    6     10  16 USER02 $CAI00.DI.LIST
 PS   06/10/00    19    6     13   1 USER02 $CAI00.LOG.LIST
 PS   05/19/00    40   10     30   4 USER02 $CAI00.TV.DATA
 PS   05/19/00    11    1     10   2 USER02 $CAI00.VG.LIST

 |    |           |    |      |   | |      |

 PS               38    0     38   1 USER02 $CAI02.VTLST1
 PS   06/11/00    20   14      6   1 USER02 $CAI03.DG.ASM
 PS   05/19/00     8    1      7   1 USER02 $CAI04.EDT.BKUP
 PS   05/29/00    11    2      9   2 USER02 $CAI04.WTO.LIST
 PS   06/03/00   180   22    158   1 USER02 $CAI05.MAN.TEXT
 PS   06/11/00     8    1      7   1 USER02 $CAI05.EDT.BKUP
                 744  273    471  46        TOTAL
```

It also generates a $RSTRANS file containing these commands:

```
$SM DSNAME('$CAI00.$AI.LIST') RLSE
$SM DSNAME('$CAI00.D.DATA')   RLSE
$SM DSNAME('$CAI00.DI.LIST')  RLSE
$SM DSNAME('$CAI00.LOG.LIST') RLSE
$SM DSNAME('$CAI00.TV.DATA')  RLSE
$SM DSNAME('$CAI00.VG.LIST')  RLSE
|   |       |                 |
$SM DSNAME('$CAI02.VTLST1')   RLSE
$SM DSNAME('$CAI03.DG.ASM')   RLSE
$SM DSNAME('$CAI04.EDT.BACKUP') RLSE
$SM DSNAME('$CAI04.WTO.LIST') RLSE
$SM DSNAME('$CAI05.MAN.TEXT') RLSE
$SM DSNAME('$CAI05.EDT.BKUP') RLSE
```

Example 1.2 Building a Command with $SMRLSE BLIST

In this example, RSVP has a built-in action list and command defined in the $RSUSER CSECT. This command and subsequent step issue the same commands. The primary keyword is BLIST.

```
//GENTSO    EXEC ASM2CMDU
//$RSOUT    DD SYSOUT=*
//$RSTRANS  DD DSN=&&TSOSM,DISP=(,PASS),
//             UNIT=SYSDA,SPACE=(TRK,(10,5))
//SYSIN     DD *
$RSVP VOL(USER) IF(DSORG EQ PS) AND1(UNUSED GT 5) TRK -
PRINT(DEL DSCOUNT)                                    -
BLIST($SMRLSE)
/*
//TSOSM     EXEC ASM2CMDU
//SYSIN     DD DSN=&&TSOSM,DISP=(OLD,DELETE)
```

The report generated lists the data sets to have space released:

```
$RS0A03 RSVP VERSION 1.0   $RSVP VOL(USER)          PAGE  1
DSORG LSTUS    ALLOC USED UNUSED EXT VOLUME DSNAME

 PS   06/09/00    57   40     17   3 USER02 $CAI00.$AI.LIST
 PS   05/19/00    60   42     18   2 USER02 $CAI00.D.DATA
 PS   05/27/00    16    6     10  16 USER02 $CAI00.DI.LIST
 PS   06/10/00    19    6     13   1 USER02 $CAI00.LOG.LIST
 PS   05/19/00    40   10     30   4 USER02 $CAI00.TV.DATA
 PS   05/19/00    11    1     10   2 USER02 $CAI00.VG.LIST

 |    |           |    |      |   | |      |

 PS              38    0     38   1 USER02 $CAI02.VTLST1
 PS   06/11/00    20   14      6   1 USER02 $CAI03.DG.ASM
 PS   05/19/00     8    1      7   1 USER02 $CAI04.EDT.BKUP
 PS   05/29/00    11    2      9   2 USER02 $CAI04.WTO.LIST
 PS   06/03/00   180   22    158   1 USER02 $CAI05.MAN.TEXT
 PS   06/11/00     8    1      7   1 USER02 $CAI05.EDT.BKUP
                744  273    471  46        TOTAL
```

The field list specified in BLIST ($SMRLSE) generates a $RSTRANS file using action fields contained in the user CSECT $RSUSER.

```
$SM DSNAME('$CAI00.$AI.LIST') VOL(USER02) RLSE
$SM DSNAME('$CAI00.D.DATA')   VOL(USER02) RLSE
$SM DSNAME('$CAI00.DI.LIST')  VOL(USER02) RLSE
$SM DSNAME('$CAI00.LOG.LIST') VOL(USER02) RLSE
$SM DSNAME('$CAI00.TV.DATA')  VOL(USER02) RLSE
$SM DSNAME('$CAI00.VG.LIST')  VOL(USER02) RLSE
|   |       |               |           |
$SM DSNAME('$CAI02.VTLST1')   VOL(USER02) RLSE
$SM DSNAME('$CAI03.DG.ASM')   VOL(USER02) RLSE
$SM DSNAME('$CAI04.EDT.BKUP') VOL(USER02) RLSE
$SM DSNAME('$CAI04.WTO.LIST') VOL(USER02) RLSE
$SM DSNAME('$CAI05.MAN.TEXT') VOL(USER02) RLSE
$SM DSNAME('$CAI05.EDT.BKUP') VOL(USER02) RLSE
```

3.4.1.2 Example 2 - Generating $SM Commands to Release, Compress

Example 2.1 Building a Command with ACTION1 BLIST

Although it may be necessary to compress large partitioned data sets and release extra space, it may also be necessary to maintain a small portion of extra space on a PDS to allow updating of members.

```
//GENTSO    EXEC ASM2CMDU
//$RSOUT    DD SYSOUT=*
//$RSTRANS  DD DSN=&&TSOSM,DISP=(,PASS),
//             UNIT=SYSDA,SPACE=(TRK,(10,5))
//SYSIN     DD *
$RSVP VOL(USER) IF(DSORG EQ PO) AND1(LSTUS GT *-15)        -
AND2(ALLOC GT 20) AND3(UNUSED LT 5) TRK                    -
PRINT(DEL DSCOUNT)                                         -
ACTION('$SM DSNAME(''') COM(''') COMPRESS RLSE SAVE(10)')  -
BLIST(ACTION1)
/*
//TSOSM     EXEC ASM2CMDU
//SYSIN     DD DSN=&&TSOSM,DISP=(OLD,DELETE)
```

This command and subsequent step issue the necessary commands to compress selected PDSs and release unused space, saving 10 tracks for expansion. The primary keywords are ACTION, COMMENT, and BLIST. The field list specified in BLIST (ACTION1) generates a $RSTRANS file with fields ACTION, DSNAME, and COMMENT in that order. The ACTION and COMMENT fields pass the proper formats on to the $RSTRANS.

The report lists the data sets to have space released.

```
$RS0A03 RSVP VERSION 1.0   $RSVP VOL(USER)         PAGE  1
DSORG LSTUS    ALLOC USED UNUSED EXT VOLUME DSNAME

 PO   06/17/00    26   23      3   5 USER01 $CAI01.$RS.LOAD
 PO   06/17/00    41   40      1   1 USER02 $CAI03.C.AS.AS
 PO   06/15/00    25   22      3   3 USER01 $CAI04.MIG.LOAD
                  92   85      7   9        TOTAL
```

And the necessary $SM commands are generated:

```
$SM DSNAME('$CAI01.$RS.LOAD') COMPRESS RLSE SAVE(10)
$SM DSNAME('$CAI03.C.AS.AS') COMPRESS RLSE SAVE(10)
$SM DSNAME('$CAI04.MIG.LOAD') COMPRESS RLSE SAVE(10)
```

Example 2.2 Building a Command with $SMCOMP BLIST

This example uses built-in commands and actions to accomplish the same thing as Example 1. However in this case, it is actually more desirable since the built-in command actually generates a SAVE value equal to 10 percent of the prior allocation. You can change this value in the $RSEXT1 CSECT (see RSVP User Exits on page 5-4).

```
//GENTSO    EXEC ASM2CMDU
//$RSOUT    DD SYSOUT=*
//$RSTRANS  DD DSN=&&TSOSM,DISP=(,PASS),
//             UNIT=SYSDA,SPACE=(TRK,(10,5))
//SYSIN     DD *
$RSVP VOL(USER) IF(DSORG EQ PO) AND1(LSTUS GT *-15) -
AND2(ALLOC GT 20) AND3(UNUSED LT 5) TRK             -
PRINT(DEL DSCOUNT)                                  -
BLIST($SMCOMP)
/*
//TSOSM     EXEC ASM2CMDU
//SYSIN     DD DSN=&&TSOSM,DISP=(OLD,DELETE)
```

This command and subsequent step issue the same report. The primary keywords are ACTION and BLIST. The field list specified in BLIST ($SMCOMP) generates a $RSTRANS file using action fields contained in the user CSECT $RSUSER.

```
$RS0A03 RSVP VERSION 1.0   $RSVP VOL(USER)        PAGE  1
DSORG LSTUS    ALLOC USED UNUSED EXT VOLUME DSNAME

 PO   06/17/00    26   23      3   5 USER01 $CAI01.$RS.LOAD
 PO   06/17/00    41   40      1   1 USER02 $CAI03.C.AS.AS
 PO   06/15/00    25   22      3   3 USER01 $CAI04.MIG.LOAD
                  92   85      7   9        TOTAL
```

But the command generates different $SM commands:

```
$SM DSNAME('$CAI01.$RS.LOAD') VOL(USER01) COMPRESS RLSE PCTSAVE(10)
$SM DSNAME('$CAI03.C.AS.AS') VOL(USER02) COMPRESS RLSE PCTSAVE(10)
$SM DSNAME('$CAI04.MIG.LOAD') VOL(USER01) COMPRESS RLSE PCTSAVE(10)
```

3.4.1.3 Example 3 - Generating TSO DELETE Commands

You can generate TSO delete statements to eliminate short-lived data sets such as TSO LIST-type. In this example, LIST type data sets are allowed to remain for two days for a user's retrieval, but after this time they are considered outdated and can be deleted.

```
//TSOLIST  EXEC ASM2CMDU
//$RSTRANS DD DSN=&&TSODEL,DISP=(,PASS),
//            UNIT=SYSDA,SPACE=(TRK,(10,5))
//SYSUDUMP DD SYSOUT=A
//SYSIN    DD *
$RSVP VOL(USER) PRINT(DEL (DSCOUNT DSORG))       -
LASTLEVEL(LIST LINKLIST OUTLIST TESTLIST)        -
IF(CREDT LT *-2) ACTION(' DEL ''') COMMENT('''') -
BLIST(ACTION1)
/*
//*
//TSODEL   EXEC PGM=IKJEFT01,DYNAMNBR=50,TIME=(,30)
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD DSN=&&TSODEL,DISP=(OLD,DELETE)
```

This command and subsequent step issue the necessary commands to delete TSO list data sets. The primary keywords are ACTION, COMMENT, and BLIST. The field list specified in BLIST (ACTION1) generates a $RSTRANS file with fields ACTION, DSNAME, and COMMENT in that order. The ACTION and COMMENT fields pass the proper formats on to the $RSTRANS.

The report lists the data sets to have space released.

```
$RS0A03 RSVP VERSION 1.0   $RSVP VOL(USER)        PAGE  1
LSTUS    ALLOC USED UNUSED EXT VOLUME DSNAME

06/09/00  1087  763    324   3 USER02 $CAI00.$AI.LIST
06/09/00   610  610      0   2 USER02 $CAI00.$DA.LIST
05/19/00    19   19      0   1 USER02 $CAI00.DI.TESTLIST
05/19/00    19   19      0   1 USER02 $CAI00.ENT.TESTLIST
06/08/00   763  763      0   3 USER02 $CAI00.NAI.LIST
06/10/00   362  114    248   1 USER02 $CAI00.SPFLOG0.LIST
|          |    |      |    | |      |
05/19/00    38   19     19   1 USER02 $CAI04.SPFTEMP1.LIST
05/29/00   210   38    172   2 USER02 $CAI04.WTO.TESTLIST
           572  572      0   1 USER02 $CAI05.DUMP.LIST
06/02/00    19   19      0   1 USER02 $CAI05.SPFLOG1.LIST
05/19/00    76   76      0   1 USER02 $CAI99.GENCMD.LIST
05/19/00  1907  477   1430   1 USER02 $CAI99.VSAMPLUS.LIST
         10469 7647   2822  37        TOTAL
```

The delete statements are also generated:

```
DEL '$CAI00.$AI.LIST'
DEL '$CAI00.$DA.LIST'
DEL '$CAI00.DI.TESTLIST'
DEL '$CAI00.ENT.TESTLIST'
DEL '$CAI00.NAI.LIST'
DEL '$CAI00.SPFLOG0.LIST'
|    |
DEL '$CAI04.SPFTEMP1.LIST'
DEL '$CAI04.WTO.TESTLIST'
DEL '$CAI05.DUMP.LIST'
DEL '$CAI05.SPFLOG1.LIST'
DEL '$CAI99.GENCMD.LIST'
DEL '$CAI99.VSAMPLUS.LIST'
```

3.4.1.4 Example 4 - One-Step CLIST Technique

Another method eliminates the two-step process associated with preprocessing. You simply put the DELETE commands into a data set and execute it as a CLIST. In this case RSVP is invoked under the TSO batch TMP, which can also process CLIST input.

```
//TSOCLIST EXEC PGM=IKJEFT01,DYNAMNBR=50,TIME=(,30)
//*
//*        GENERATE TSO DELETE CARDS ....
//*        THEN EXECUTE AS A CLIST
//*
//STEPLIB  DD DISP=SHR,DSN=$CAI01.$RS.LOAD
//SYSTSPRT DD SYSOUT=*
//$RSOUT   DD SYSOUT=*
//$RSTRANS DD DSN=$CAI01.DEL.CLIST,DISP=OLD
//SYSTSIN  DD *
$RSVP VOL(USER) PRINT(DEL DSCOUNT)               -
LASTLEVEL(LIST LINKLIST OUTLIST TESTLIST)        -
IF(CREDT LT *-2) ACTION(' DEL ''') COMMENT('''') -
BLIST(ACTION1)
//*
EX '$CAI01.DEL.CLIST'
```

3.4.1.5 Example 5 - Reporting SMS Information from the IPC

The following JCL searches the IPC for SMS controlled data sets with certain high-level qualifiers that were archived and have the Storage Class of BATCH. It can list them and also generate CA-ASM2 commands to reload all of them.

```
//GEN$RA  EXEC  ASM2CMDU
//$RSOUT   DD   SYSOUT=*
//$RSTRANS DD   DSN=&&TSORA,DISP=(,PASS),
//              UNIT=SYSDA,SPACE=(TRK,(10,5)
//SYSIN    DD   *
$RSVP  ASM2CAT LEVEL(CN9000 BUEJO01 BAIMI02)   -
       IF(STRCLASS EQ BATCH) AND1(UNLV# EQ 0)  -
       AND2(IPCUTYPE EQ ARCH) -
       PRINT(NEW (MGTCLASS DATCLASS STRCLASS UNLVOL UNLFSEQ DSN)) -
       BLIST(ACTION1)  SMSNMLEN(9) -
       ACTION('$RA DSNAME(''')  COMMENT(''')')
/*
//TSO$RA  EXEC  ASM2CMDU
//SYSIN    DD   DS&&TSORA,DISP=(OLD,DELETE)
```

The initial $RSVP command selects only data sets beginning with the high-level qualifiers CN9000, BUEJO01, and BAIMI02. Only the most recently archived data set versions (UNLV# EQ 0 and IPCUTYPE EQ ARCH) are selected. A transaction file containing $RA commands is generated and eventually executed in the TSO$RA step.

The report generated during the GEN$RA step is shown here.

```
$RS0A03 RSVP VERSION 1.0       $RSVP   ASM2CAT  LEVEL(*) LIST(SPACE) PRIN T(NEW (MGTCLASS DATCLASS STRCLASS UNLVOL UNLFSEQ  PAGE 1
MGTCLASS  DATCLASS  STRCLASS  UNLVOL UNLFSE DSNAME

LVL190    SEQFB     BATCH     ASF062     43 BABBI02.RMF.FIX.TEST
LVL190    SEQFB     BATCH     ASF062     41 BABBI02.RMF.TEST
LVL190    SEQFB     BATCH     ASF062     46 BABBI02.RMF.TEST2
SKPLV1    SEQTST    BATCH     ASH002      2 BUBBA01.$COPYTP.TEST1
SKPLV1    SEQTST    BATCH     ASH032      2 BUBBA01.$COPYTP.TEST1
SKPLV1    SEQTST    BATCH     ASF070    333 BUBBA01.$COPYTP.TEST1
SKPLV1    SEQTST    BATCH     ASF061      4 BUBBA01.$COPYTP.TEST2
LVL190    SEQFB     BATCH     ASF062      4 BUBBA01.$FM.NO.DATA.IN.IXR.CATALOG
LVL190    SEQFB     BATCH     ASF062      4 BUBBA01.$FM.SPANNED.TEST1
GDGMGT    DATGDG    BATCH     ASF059     32 BUBBA01.DLA.GDGTST.G0002V00
GDGMGT    DATGDG    BATCH     ASF059     33 BUBBA01.DLA.GDGTST.G0003V00
GDGMGT    DATGDG    BATCH     ASF059     23 BUBBA01.GDG.LVL05.G0001V00
GDGMGT    DATGDG    BATCH     ASF059     24 BUBBA01.GDG.LVL05.G0002V00
LVL190    SEQFB     BATCH     ASF061      5 BUBBA01.JCLCHECK.TEST@SEC
LVL190    SEQFB     BATCH     ASF061      7 BUBBA01.MULTIVOL.TEST
LVL190    SEQFB     BATCH     ASF061     17 BUBBA01.POST.IXR1Y
LVL190    SEQFB     BATCH     ASF061     18 BUBBA01.POST.IXR2Y
LVL190    SEQFB     BATCH     ASF061     19 BUBBA01.POST.IXR3Y
LVL190    SEQFB     BATCH     ASF061     20 BUBBA01.POST.IXR4N
LVL190    SEQFB     BATCH     ASF061     21 BUBBA01.POST.IXR5Y
LVL190    SEQFB     BATCH     ASF061     22 BUBBA01.POST.IXR6N
LVL190    SEQFB     BATCH     ASF062     54 BUBBA01.TEST.DATASETA
LVL190    SEQFB     BATCH     ASF062     55 BUBBA01.TEST.DATASETB
LVL190    SEQFB     BATCH     ASF062     56 BUBBA01.TEST.DATASETC
LVL190    SEQFB     BATCH     ASF062     48 BUBBA01.TEST.DATASET1
LVL190    SEQFB     BATCH     ASF062     49 BUBBA01.TEST.DATASET2
LVL190    SEQFB     BATCH     ASF062     50 BUBBA01.TEST.DATASET3
LVL190    SEQFB     BATCH     ASF062     51 BUBBA01.TEST.DATASET4
LVL190    SEQFB     BATCH     ASF062     52 BUBBA01.TEST.DATASET5
LVL190    SEQFB     BATCH     ASF062     53 CN9000.DLA302.ARCH.CATALOG
LVL190    SEQFB     BATCH     ASF051      4 CN9000.DLA303.ARCH.CATALOG
LVL190    SEQFB     BATCH     ASF059      9 CN9000.QC.$QM1PS
LVL190    SEQFB     BATCH     ASF059     11 CN9000.QC.$QM1PSFB
LVL190    SEQFB     BATCH     ASF059     10 CN9000.QC.$QM1PSV
LVL190    SEQFB     BATCH     ASF059      8 CN9000.QC.NEW5DA
LVL190    SEQFB     BATCH     ASF059      5 CN9000.QC.NEW5PS
LVL190    SEQFB     BATCH     ASF059      7 CN9000.QC.NEW5PSFB
LVL190    SEQFB     BATCH     ASF059      6 CN9000.QC.NEW5PSV
LVL190    SEQFB     BATCH     ASF059      3 CN9000.QC.TST5DA
```

Field Descriptions

MGTCLASS SMS management class name associated with the data set.

DATCLASS SMS data class name associated with the data set.

STRCLASS SMS storage class name associated with the data set.

UNLVOL Volume serial number to which the data set was unloaded.

UNLFSE File sequence number of the unload volume.

DSNAME Data set name.

3.4.2 Archive and Backup Queues Example

In this example, the SUBCOMMAND keyword generates $AR commands that place the data set names directly into the archive queue. Data sets are selected using the LIKE keyword.

```
$RSVP VOL(USER01) LIKE($CAI01.-) SUBCOMMAND($AR) -
COMMENT('TESTRSVP')  RETPD(88)
```

This command generates the following report:

```
DSCOUNT DSORG LSTUS    ALLOC  USED UNUSED EXT  VOLUME DSNAME  PAGE 1

        PO    4/26/00    720  384     336   1  USER01 $CAI01.DATA1
        PO    4/27/00   1007  384     623   1  USER01 $CAI01.OBJ
        PS    5/07/00     48   48       0   1  USER01 $CAI01.RAORD
        PS    5/07/00     48    0      48   1  USER01 $CAI01.RBORD
        PS    5/07/00     48   48       0   1  USER01 $CAI01.RSTRANS
     5                  1871  864    1007   5
```

$AR commands are created for each of the five data sets and the data set names are placed in the archive queue. The $QM command verifies this:

```
$QM DSN(-.-) PRINT(DSN COMMENT)
```

And the command also generates this output:

```
$QM0100I  $QM ACTION  (LIST)
$QM0100I      COMMAND ($AR)
$QM0100I      DSNAME  (-.-)
$QM0100I      PRINT   (DSNAME,
$QM0100I               COMMENT)
$QM0100I      SORT    (DSNAME)

DSNAME               COMMENT
$CAI01.DATA1         TESTRSVP
$CAI01.OBJ           TESTRSVP
$CAI01.RAORD         TESTRSVP
$CAI01.RBORD         TESTRSVP
$CAI01.RSTRANS       TESTRSVP
$QM0400I QUEUE STATISTICS: $LISTED= 5  #USED = 5  #FREE = 188
```

3.4.3 Unregistered Data Sets Examples

3.4.3.1 Example 1 - Delete Unregistered Data Sets

This example assumes that the CA-ASM2 Open Modification has been installed. In this example, only valid user IDs (high-level indexes) are registered instead of complete data set names. In the sample JCL, if the high-level index of some data set, as recorded in the VTOC of a volume, does not match a high-level index in the CA-ASM2 registration table, the ACCOUNT for that data set is filled in as UNKNOWN and the data set is selected for deleting in the second step.

The sample JCL for this example is:

```
//TSOBILL  EXEC ASM2CMDU
//$RSEXT2R DD DSN=$CAI01.$RS.EXT2S,DISP=SHR
//$RSTRANS DD DSN=&&BLTRANS,DISP=(,PASS),
//            UNIT=SYSDA,SPACE=(TRK,(10,5))
//SYSUDUMP DD SYSOUT=A
//SYSIN    DD *
$RSVP VOL(USER) NOPRINT SORT(DSNAME)BLIST(BILLLIST)
/*
//GENTSO   EXEC ASM2CMDU
//*
//*        GENERATE TSO DELETE CARDS ...
//*
//$RSTRANS DD DSN=&&TSODEL,DISP=(,PASS),
//            UNIT=SYSDA,SPACE=(TRK,(10,5))
//BILL     DD DSN=&&BLTRANS,DISP=(OLD,PASS)
//SYSIN    DD *
$RSVP INBILL(BILL) INBLIST(BILLLIST)         -
IF(ACCOUNT EQ UNKNOWN)                       -
ACTION(' DEL ''') COMMENT('''')  BLIST(ACTION1) -
PRINT(NEW (DSCOUNT KBALLOC VOLUME DSNAME)) NOSORT
/*
//*
//TSODEL   EXEC PGM=IKJEFT01,DYNAMNBR=50,TIME=(,30)
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD DSN=&&TSODEL,DISP=(OLD,DELETE)
```

This command and subsequent step issue the necessary commands to delete unregistered data sets. The first step processes the DASD volumes matching the registration file to the data sets. The NOPRINT keyword causes RSVP to default to NOSORT to improve performance, but the exit ($RSEXT2), which matches data set names to registration records, expects the names to be in order. Therefore, you must include the keyword SORT(DSNAME).

The second step processes the output of the first, looking for unregistered data sets. The keywords can be looked at in two groups. The first group is INBILL and INBLIST. INBILL specifies the DDNAME of a file from a previous $RSVP command, and INBLIST specifies the field list used in the previous run for $RSTRANS (that is, the BLIST used in the previous run). The second group of keywords are ACTION, COMMENT, and BLIST. The field list specified in BLIST (ACTION1) generates a $RSTRANS file with fields ACTION, DSNAME, and

COMMENT in that order. The ACTION and COMMENT fields pass the proper formats on to the $RSTRANS.

This report lists the data sets to be deleted:

```
$RS0A03 RSVP VERSION 1.0   $RSVP INBILL(BILL) PAGE  1
DSCOUNT  KBALLOC    VOLUME    DSNAME

             572    SYS500    $CAI05.$RS.DATA
             572    SYS500    $CAI05.ACCT.ASM
            1144    SYS500    $CAI05.ASM2MAN.TEXT
              38    SYS500    $CAI05.ASM2MISC.TEXT

               |    |         |       |

             114    SYS500    $CAI05.VSAM.LIST
             305    USER01    $CAI98.L703.VTOC.DATA
             305    USER01    $CAI98.T703.VTOC.CARDS
     21    15157
```

And it generates the correct TSO DELETE commands:

```
DEL '$CAI05.$RS.DATA'
DEL '$CAI05.ACCT.ASM'
DEL '$CAI05.ASM2MAN.TEXT'
DEL '$CAI05.ASM2MISC.TEXT'
|   |       |
DEL '$CAI05.VSAM.LIST'
DEL '$CAI98.L703.VTOC.DATA'
DEL '$CAI98.T703.VTOC.CARDS'
```

The registration file ($RSEXT2R) looks like this:

```
L$CAI00                                  DICK
L$CAI01                                  TOM
L$CAI02                                  MARGE
L$CAI03                                  JOE
L$CAI04                                  LEE
L$CAI99                                  PROD
```

The format of the $RSEXT2R file is shown on $RSEXT2R on page 4-8 in the "DASD Billing" chapter.

The intermediate file resembles this sample:

```
06/17/003390USER01$CAI00.V.CLIST           DICK      19   0
06/17/003390USER01$CAI01.$RS.VOLTM         TOM       38   0
06/17/003390USER02$CAI02.ZAPSCAN.OBJ       MARGE     19   0
06/17/003390SYS500$CAI03.ARCH.DATA         JOE     2860   0
06/17/003390USER02$CAI04.DUMMY             LEE       19   0
06/17/003390SYS500$CAI05.$RS.DATA          UNKNOWN  572   0
06/17/003390SYS500$CAI05.ACCT.ASM          UNKNOWN  572   0
06/17/003390SYS500$CAI05.ASM2MAN.TEXT      UNKNOWN 1144   0
06/17/003390SYS500$CAI05.ASM2MISC.TEXT     UNKNOWN   38   0
06/17/003390SYS500$CAI05.VSAM.LIST         UNKNOWN  114   0
06/17/003390USER01$CAI98.L703.VTOC.DATA    UNKNOWN  305   0
06/17/003390USER01$CAI98.T703.VTOC.CARDS   UNKNOWN  305   0
06/17/003390SYS500$CAI99.ASM2GEN.CNTL      PROD     420   0
06/17/003390SYS500$CAI99.ZAP.DATA          PROD      19   0
```

It may not be possible to rely on TSO DELETE commands to delete data sets since they may not always be cataloged. If this is the case, you can generate IEHPROGM control statements instead:

```
//TSOBILL  EXEC ASM2CMDU
//$RSEXT2R DD DSN=$CAI01.$RS.EXT2S,DISP=SHR
//$RSTRANS DD DSN=&&BLTRANS,DISP=(MOD,PASS),
//            UNIT=SYSDA,SPACE=(TRK,(10,5))
//SYSUDUMP DD SYSOUT=A
//SYSIN    DD *
$RSVP VOL(USER) NOPRINT SORT(DSNAME) BLIST(BILLLIST)
/*
//*
//GENPROGM EXEC ASM2CMDU
//*
//*        GENERATE IEHPROGM CONTROL CARDS
//*
//SYSUDUMP DD SYSOUT=A
//$RSTRANS DD DSN=&&PROGM,DISP=(MOD,PASS),
//            UNIT=SYSDA,SPACE=(TRK,(10,5))
//BILL     DD DSN=&&BLTRANS,DISP=(OLD,PASS)
//SYSIN    DD *
$RSVP INBILL(BILL) INBLIST(BILLLIST) BLIST(ACTION2)     -
ACTION(' SCRATCH VOL=3390=') COMMENT(',DSNAME=')        -
PR(NEW (KBALLOC VOLUME DSNAME)) IF(ACCOUNT EQ UNKNOWN)  -
USERCSECT($RSUSER8) NOSORT
$RSVP INBILL(BILL) INBLIST(BILLLIST) BLIST(ACTION1)     -
ACTION(' UNCATLG DSNAME=') COMMENT(' ')                 -
NOPRINT(NEW (DSCOUNT KBALLOC)) IF(ACCOUNT EQ UNKNOWN)   -
USERCSECT($RSUSER8) NOSORT
/*
//*
//IEHPROGM EXEC PGM=IEHPROGM
//DISK01   DD UNIT=3390,VOL=SER=USER01
//DISK02   DD UNIT=3390,VOL=SER=USER01
//SYSIN    DD DSN=&&PROGM,DISP=(OLD,DELETE)
//*
```

There are several differences in this approach:

1. In the second step, the $RSVP command must be executed twice, once for SCRATCH and once for UNCATLG.
2. Because it is executed twice, the $RSTRANS file must specify DISP=MOD or you only get the last command.
3. The $RSVP command to generate SCRATCH statements uses BLIST(ACTION2), which generates a $RSTRANS record with fields ACTION, VOLUME, COMMENT, and DSNAME.
4. Since IEHPROGM can only process 80-byte card images, the USERCSECT keyword is used to select an alternate user CSECT. $RSUSER8 is a duplicate of $RSUSER except that it specifies record length of 80 and block size of 6160 on $RSTRANS.

The report is the same; the second execution specifies NOPRINT to eliminate duplication.

```
$RS0A03 RSVP VERSION 1.0   $RSVP INBILL(BILL) PAGE  1
 KBALLOC   VOLUME   DSNAME

     572   SYS500   $CAI05.$RS.DATA
     572   SYS500   $CAI05.ACCT.ASM
    1144   SYS500   $CAI05.ASM2MAN.TEXT
      38   SYS500   $CAI05.ASM2MISC.TEXT
       |   |    |   |       |
     114   SYS500   $CAI05.VSAM.LIST
     305   USER01   $CAI98.L703.VTOC.DATA
     305   USER01   $CAI98.T703.VTOC.CARDS
   15157
```

And the following IEHPROGM control statements are generated:

```
SCRATCH VOL=3390=SYS500,DSNAME=$CAI05.$RS.DATA
SCRATCH VOL=3390=SYS500,DSNAME=$CAI05.ACCT.ASM
SCRATCH VOL=3390=SYS500,DSNAME=$CAI05.ASM2MAN.TEXT
SCRATCH VOL=3390=SYS500,DSNAME=$CAI05.ASM2MISC.TEXT
|       |   |    |       |       |
SCRATCH VOL=3390=SYS500,DSNAME=$CAI05.VSAM.LIST
SCRATCH VOL=3390=USER01,DSNAME=$CAI98.L703.VTOC.DATA
SCRATCH VOL=3390=USER01,DSNAME=$CAI98.T703.VTOC.CARDS
UNCATLG DSNAME=$CAI05.$RS.DATA
UNCATLG DSNAME=$CAI05.ACCT.ASM
UNCATLG DSNAME=$CAI05.ASM2MAN.TEXT
UNCATLG DSNAME=$CAI05.ASM2MISC.TEXT
|       |       |
UNCATLG DSNAME=$CAI05.VSAM.LIST
UNCATLG DSNAME=$CAI98.L703.VTOC.DATA
UNCATLG DSNAME=$CAI98.T703.VTOC.CARDS
```

3.4.3.2 Example 2 - Generating IEHPROGM SCRATCH/UNCATLG Statements

It may be necessary to delete data sets from a volume but impossible to use the TSO DELETE command since it is not known whether a data set is cataloged. In this case, it is beneficial to generate IEHPROGM control statements to scratch and uncatalog the data sets. This example deletes any data set on a specific storage volume if it has not been used for the last six months (180 days).

```
//TSOBILL  EXEC ASM2CMDU
//SYSUDUMP DD SYSOUT=A
//$RSTRANS DD DSN=&&PROGM,DISP=(MOD,PASS),
//            UNIT=SYSDA,SPACE=(TRK,(10,5))
//SYSIN    DD *
$RSVP VOL(USER) IF(LSTUS LT *-180)                -
PRINT(DEL DSCOUNT)                                -
ACTION(' SCRATCH VOL=3390=') COMMENT(',DSNAME=')   -
USERCSECT($RSUSER8) BLIST(ACTION2)
$RSVP VOL(USER) IF(LSTUS LT *-180) AND1(CAT EQ C)  -
NOPRINT                                           -
ACTION(' UNCATLG DSNAME=') COMM(' ')              -
USERCSECT($RSUSER8) BLIST(ACTION1)
/*
//IEHPROGM EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=*
//DISK01   DD UNIT=3390,VOL=SER=USER01,DISP=OLD
//DISK02   DD UNIT=3390,VOL=SER=USER01,DISP=OLD
//SYSIN    DD DSN=&&PROGM,DISP=(OLD,DELETE)
```

These commands and subsequent step issue the necessary IEHPROGM control statements to scratch and uncatalog the selected data sets. The primary keywords are ACTION, COMMENT, and BLIST.

The $RSVP command to generate SCRATCH statements uses BLIST(ACTION2), which generates a $RSTRANS record of fields ACTION, VOLUME, COMMENT, DSNAME. The command to generate UNCATLG statements uses BLIST (ACTION1), which generates a $RSTRANS file with fields ACTION, DSNAME, and COMMENT in that order. The ACTION and COMMENT fields are used to pass the proper formats on to the $RSTRANS. Since IEHPROGM can only take 80-byte card images, the keyword USERCSECT is used to reference a CSECT similar to $RSUSER except that the $RSTRANS file is specified as record length 80, block size 6160.

The command to generate the UNCATLG statements has AND1(CAT EQ C) to avoid generating an UNCATLG for a data set that is cataloged to a different volume.

The report lists the data sets to be deleted as shown in the following example.

```
$RS0A03 RSVP VERSION 1.0   $RSVP VOL(USER)       PAGE  1
DSORG LSTUS    ALLOC USED UNUSED EXT VOLUME DSNAME

 PO   12/05/99    19   19      0   1 USER02 $CAI00.CDG.ASM
 PS   01/19/00   725    0    725   1 USER02 $CAI02.VTLST1
 PO   06/30/99   953   19    934   1 USER02 $CAI03.DG.ASM
 PS   09/23/99    57   57      0   1 USER02 $CAI03.TEMP.WYL
 PS   01/13/00    19   19      0   1 USER02 $CAI04.DUMMY
 PS   11/04/99     0    0      0   0 USER01 $CAI04.USER01
 PS   01/28/00     0    0      0   0 USER02 $CAI04.USER02
 PO   10/13/99   572  572      0   1 USER02 $CAI05.D.LIST
               2345  686   1659   6        TOTAL
```

It also generates the correct control statements:

```
SCRATCH VOL=3390=USER02,DSNAME=$CAI00.CDG.ASM
SCRATCH VOL=3390=USER02,DSNAME=$CAI02.VTLST1
SCRATCH VOL=3390=USER02,DSNAME=$CAI03.DG.ASM
SCRATCH VOL=3390=USER02,DSNAME=$CAI03.TEMP.WYL
SCRATCH VOL=3390=USER02,DSNAME=$CAI04.DUMMY
SCRATCH VOL=3390=USER01,DSNAME=$CAI04.USER01
SCRATCH VOL=3390=USER02,DSNAME=$CAI04.USER02
SCRATCH VOL=3390=USER02,DSNAME=$CAI05.D.LIST
UNCATLG DSNAME=$CAI02.VTLST1
UNCATLG DSNAME=$CAI03.DG.ASM
UNCATLG DSNAME=$CAI03.TEMP.WYL
UNCATLG DSNAME=$CAI04.USER01
UNCATLG DSNAME=$CAI04.USER02
UNCATLG DSNAME=$CAI05.D.LIST
```

3.4.3.3 Example 3 - Generating IEHPROGM SCRATCH Statements

In this example, the selection is limited to data sets that are either uncataloged, or miscataloged (cataloged to a different volume). There is also a selection on create date at least two days old to avoid problems on data sets currently being created.

```
//TSOSEL   EXEC ASM2CMDU
//$RSTRANS DD DSN=&&PROGM,DISP=(,PASS),
//            UNIT=SYSDA,SPACE=(TRK,(10,5))
//SYSIN    DD *
$RSVP VOL(USER) BLIST(SCRATCH)                        -
IF(CAT EQ N) OR1(CAT EQ W) AND2(CREDT LT *-1)         -
PRINT(NEW (KBALLOC VOLUME DSNAME)) USERCSECT($RSUSER8)
/*
//IEHPROGM EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=*
//DISK01   DD UNIT=3390,VOL=SER=USER01,DISP=OLD
//DISK02   DD UNIT=3390,VOL=SER=USER01,DISP=OLD
//SYSIN    DD DSN=&&PROGM,DISP=(OLD,DELETE)
```

This command and subsequent step issue the necessary IEHPROGM control statements to scratch uncataloged data sets. The keyword USERCSECT is used to specify a user CSECT with different LRECL and BLKSIZE on the $RSTRANS data set.

This report lists the data sets to be scratched.

```
$RS0A03 RSVP VERSION 1.0   $RSVP VOL(USER) PAGE  1
 KBALLOC   VOLUME   CAT   DSNAME

      19   USER02    W    $CAI01.$RS.ASM
      19   USER02    N    $CAI01.$RS.HIST
      38   USER02    N    $CAI01.TEMP
      95   USER02    W    $CAI04.IOPROGM.LOAD
     171                   TOTAL
```

And it generates the proper IEHPROGM control statements:

```
SCRATCH VOL=3390=USER02,DSNAME=$CAI01.$RS.ASM
SCRATCH VOL=3390=USER02,DSNAME=$CAI01.$RS.HIST
SCRATCH VOL=3390=USER02,DSNAME=$CAI01.TEMP
SCRATCH VOL=3390=USER02,DSNAME=$CAI04.IOPROGM.LOAD
```

Chapter 4. DASD Billing

RSVP provides a valuable configuration management function: resource accounting. As a stand-alone tool, it maintains data for DASD billing systems. With RSVP, you can maintain different billing rates for different volumes or device types.

- RSVP services chargebacks for DASD usage by issuing user-accounting statements. You can apply factors to charges on an account basis, and easily update them as required.
- RSVP can also be used as a front-end processor to interface with other billing systems such as CA-JARS. SMF-like billing records are generated for input into other systems
- A distinct advantage of this billing system is forecasting. It not only services billing statements for current occupancy, it also generates a billing history file that you can use to predict, by account, DASD usage and growth.

4.1 Billing

Conceptually, the DASD billing process is simple:

- Determine the amount of space allocated to the data set.
- Apply a rate factor to determine the charge.
- Determine who should be charged.
- Create a detail record for each data set that contains an account number and current charge.
- Sort detail records by account number.
- Match detail records to a history file and apply charges.
- Create a new billing history file and produce a report.

RSVP billing, both Two-Step and One-Step billing, is handled in user exits $RSEXT1 and $RSEXT2. $RSEXT1 determines the charges, and $RSEXT2 determines the account number and updates the billing history file. $RSBILL writes the billing records.

For billing purposes, RSVP uses data set registration as a way of assigning data sets to users, but instead of users, it uses the term "account." ACCOUNT is a RSVP field name. An ACCOUNT is associated with a data set in one of two places during processing. Basically, these are before the Sort phase and after the Sort phase.

In Two-Step billing, the ACCOUNT field is filled in **after** the Sort phase. Charges are computed before the sort is performed. The account field is filled in after the sort by reading records from the $RSEXT2R (DDNAME) data set (described on $RSEXT2R on page 4-8). Each record contains a data set name or data set name prefix and an account to associate with data sets matching these names or prefixes. Thus, the registration of a data set to an account is accomplished through the registration data set that has been allocated to the $RSEXT2R DDNAME. If a data set cannot be matched to an entry in the registration file, the account name is filled in with UNKNOWN. Figure 5 on 4-3 shows the Two-Step billing process.

In One-Step billing, the ACCOUNT field is filled in before the Sort phase by one of four methods. These methods are described in One-Step Billing on page 4-16. Thus, the registration of a data set to an account is accomplished by one of the four methods.

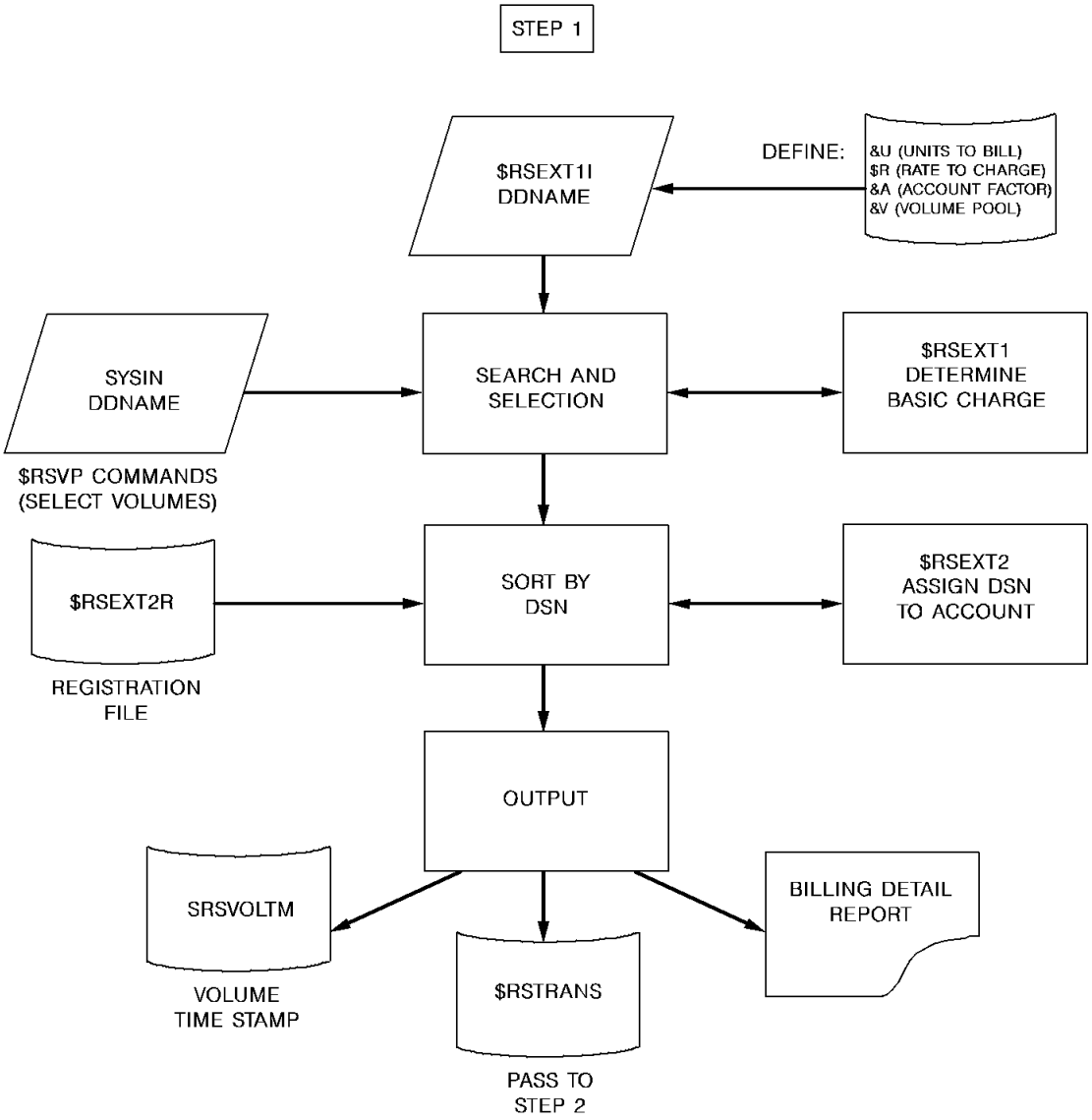


Figure 5 - RSVP Two-Step Billing (Step 1)

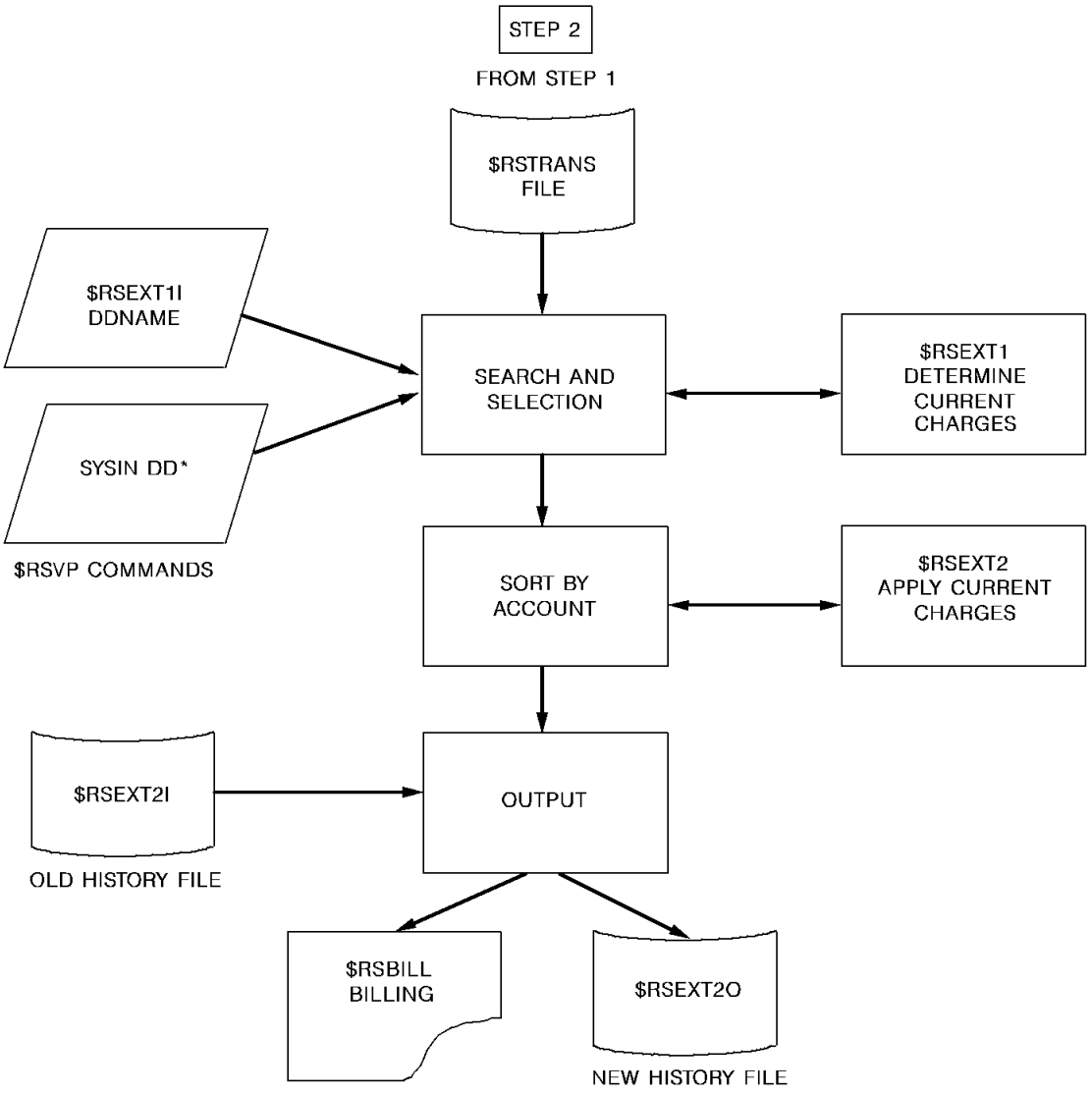
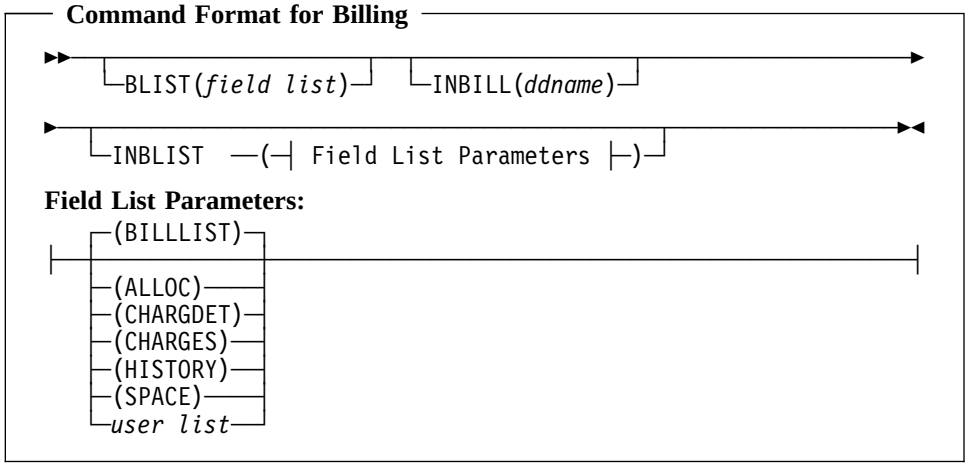


Figure 5 - RSVP Two-Step Billing (Step 2)

4.1.1 Command Format for Billing



Keyword Descriptions for Billing:

For examples using these keywords, see Two-Step Billing on page 4-11 and One-Step Billing on page 4-16.

BLIST Specifies the billing field list used for the billing transaction records going to DDNAME $RSTRANS. The default is BILLLIST.

INBILL Specifies an input billing transaction file from a previous step. This cannot reference $RSTRANS and the DD statements cannot reference the same data set.

INBLIST Specifies the field list of the input transaction file referenced in INBILL. This defaults to the list referenced in BLIST. If neither INBLIST nor BLIST are specified, it defaults to BILLLIST.

List	Fields to Print
ALLOC	DSCOUNT, ALLOC, USED, UNUSED, PCT, EXT, SECQ, SECT, DSNAME, VOLUME, DSORG, RECFM, BLKSZ, LRECL, CREDT, LSTUS
BILLLIST	DATE, UNIT, VOLUME, DSNAME, ACCOUNT, KBALLOC, KBDAYS, CHCUR
CHARGDET	ACCOUNT, DSCOUNT, KBALLOC, KBDAYS, CHCUR, CHMTD, CHYTD, UNIT, VOLUME, DSNAME
CHARGES	ACCOUNT, DSCOUNT, KBALLOC, KBDAYS, CHCUR, CHMTD, CHYTD
HISTORY	DATE, CHYTD, CHMTD, CHMONTHS, ACCOUNT
SPACE	DSCOUNT, DSORG, LSTUS, ALLOC, USED, UNUSED, EXT, VOLUME, DSNAME

4.1.2 Billing DD Statements

The following DD statements are used in a billing run:

DDNAME	Description
$RSTRANS	Billing transaction file. This can be either a fixed-block file or a variable-blocked file. A standard SMF header may preface the actual record. To get a variable-blocked and/or SMF-like file, you must change the user CSECT (see 5-9 in the "Tailoring" chapter). The default billing record format (all fields EBCDIC character, numeric fields leading zero suppressed) follows:

Columns	Description
1-8	Date of billing: MM/DD/YY.
9-16	Unit type: 3390, and so on.
17-22	VOLSER.
23-66	Data set name.
67-76	Account number.
77-84	Kilobytes allocated.
85-92	Kilobyte-days (alloc x # days).
93-100	Current charges in the form nnnnn.nn.

The variable SMF record is:

1	SMF flag.
2	SMF record ID (defined in user CSECT).
3-6	Time, binary (SMF time stamp).
7-10	Date, packed decimal (SMF time stamp).
11	Billing type SMF record.

The record format may be modified to meet your organization's requirements. See User CSECT on page 5-9 in the "Tailoring" chapter.

$RSEXT1I	Input to the Selection Exit $RSEXT1. This file defines the units to bill for, the rate to bill, and an account factor, if needed. The format of this file is:

&U Units to bill. This is followed by either:

```
KBDAYS
KBHOURS
TRKDAYS
TRKHOURS
```

The default is KBDAYS.

Whichever unit you choose to bill on **must** be referenced in either the billing list or the report list, preferably in the billing list so the units billed are available for future use. You can change the default in the user CSECT (see 5-9 in the "Tailoring: chapter).

&R Rate to charge. The format of this card is rate, unit type, and volume. Therefore, you can enter multiple rate cards. The default is .001. Some examples of rates are:

```
&R 0.00024 3390 SYS   (1)
&R 0.00012 3390       (2)
&R 0.00018 3390-1     (3)
&R 0.00048            (4)
```

Notice you do not have to enter the VOLSER in its entirety. In example (1) all 3390 volumes beginning with SYS receive a rate of 0.00024 dollars per unit. In example (2) all 3390 volumes that do not begin with SYS are charged at 0.00012 dollars per unit. In example (3) all 3390-1 volumes are charged 0.00018 dollars per unit. In example (4) any unit other than 3390 is charged 0.00048 dollars per unit.

When deciding your rates, be sure to include the control unit and the additional cost of head of string besides the pack and unit cost. RSVP can charge different rates on different volumes, if desired. For example, you can charge more for user data on system or storage volumes.

&A Account factor. The format is &A rate account. This allows different accounts to be charged differently. For example, &A 1.10 TESTACCT would cause any data set with account TESTACCT to be billed with a 10 percent surcharge (at 1.10 times the normal rate). This control statement is valid only if the account number is present in $RSEXT1, if it is being obtained from the DSCB, or if the registration file ($RSEXT2R) was used in a prior pass of the data.

The actual charge (CHCUR) is derived using the following formula:

```
space-units x rate x account-factor
```

where space-units = allocation x time-since-last-billed.

&V Volume pool. This field defines the volume pool (field VOLPOOL). The &V is followed by a volume serial or some portion thereof that is followed by a volume pool name. For example, &V SYS SYSTEMS.

An example of the input stream for Selection exit $RSEXT1 is shown on the next page.

This is an example of the input stream:

```
//$RSEXT1I  DD  *
&U KBDAYS          Bill for kilobyte-days
&R .0018 3390 SYS
&R .0012
&A .80 PROD
&V SYS SYSTEMS
&V WORK SYSDA
&V USER TSOSPACE
&V DSK PRODUCT
&E                  Exclude the following:
SYS1.              All system data sets (starting with SYS1)
SYSCTLG             CVOL catalogs
-.LIST              TSO list data sets
-.OUTLIST           TSO output list data sets
PROD***.-.DAILY     Daily production data sets (by data
*                        center standards)
PROD***.-.TEMP.-    Temporary production data sets (by data
*                        center standards)
/*
```

$RSEXT2R Input to the Post-Sort Exit $RSEXT2. This file, used as a data set register, matches a data set name to an account. It can contain records that have the full data set name or just the first part of the name. You can also specify the unit address and VOLSER.

The registration records contain the following (all fields are character format):

Columns	Description
1	L if this is a level or prefix (initial character string). D if a full data set name.
2-45	Data set name or prefix, or level.
55-end	Account number associated with this data set name.

Matching data sets takes place after the sort on dsname. If the volume and unit number are not available, the logical sort order is by data set name. The register entries must also be in data set name order, except a blank must be considered high in the collating sequence. This is necessary to allow the more specific entry to be found first.

Note: The registration file $RSEXT2R has a maximum LRECL of 120 characters. Its RECFM must be Fixed Block.

$RSEXT2I Account billing history. It is input from a prior run and the current charges are added to produce a new history going to $RSEXT2O. The history data sets contain the following data (character format):

Columns	Description
1-8	Date of this history record in MM/DD/YY format.
9-16	Year-to-date charges for account nnnnnn.nn format.
17-24	Month-to-date charges for account nnnnnn.nn format.
25-120	Charges for each month (January - December).
121-end	Account number.

$RSEXT2O History output. Contains the input from $RSEXT2I with current charges added producing a new, updated history file.

Note: The DCB attributes are not needed in the JCL with one exception, the first time billing history is generated, a dummy history input must be specified and the DCB attributes are needed as well. No history output is generated unless there is an input (even though it is a dummy data set). The DD statement for the first history run is:

```
//$RSEXT2I DD DUMMY,DCB=(RECFM=FB,LRECL=150,BLKSIZE=150)
```

$RSVOLTM Volume timestamp file. This file contains records indicating the last time billing was done on each volume. All fields are in character format to allow editing, if necessary. When billing takes place on a volume, a record is written out in the format:

Columns	Description
1-6	Volume Serial.
7-11	Date last billed.
12-15	Time last billed.
16-20	Reserved for future use.

Note: The volume timestamp file $RSVOLTM must be preallocated with the correct DCB attributes. This can be done under TSO or in batch with an IEBGENER:

```
//ALLOC    EXEC PGM=IEBGENER
//SYSPRINT  DD  SYSOUT=A
//SYSIN     DD  DUMMY
//SYSUT1    DD  DUMMY,
//              DCB=(RECFM=FB,LRECL=20,BLKSIZE=6120)
//SYSUT2    DD  UNIT=unitname,DSN=anyname,
//              DISP=(NEW,CATLG),SPACE=(TRK,1),
//              VOL=SER=volume
```

4.2 Two-Step Billing

With the RSVP exits that are supplied, two steps are necessary to complete billing. In the first step, the ACCOUNT is filled in after the sort is completed, and then in the second step, the records must be sorted into account order to maintain a billing history by account.

4.2.1 Sample JCL for Two-Step Billing

```
//BILL     EXEC ASM2CMDU
//SYSUDUMP DD SYSOUT=A
//$RSTRANS DD DSN=&&BLTRANS,DISP=(,PASS),
//            UNIT=SYSDA,SPACE=(TRK,(10,5))
//$RSEXT2R DD DSN=$CAI01.$RS.REGISTR,DISP=SHR
//$RSVOLTM DD DSN=$CAI01.$RS.VOLTM,DISP=OLD
//$RSEXT1I DD *
&U KBDAYS
&R 0.00018 3390 SYS
&R 0.00012 3390
&R 0.00024
//SYSIN    DD *
$RSVP VOL(ALL) LIST(CHARGDET) BLIST(BILLLIST) -
HEADING1('1          BILLING DETAILS, JUNE - 1998') -
PRINT(DEL (CHMTD CHYTD))
/*
//ACCOUNT  EXEC ASM2CMDU
//SYSUDUMP DD SYSOUT=A
//BILL     DD DSN=&&BLTRANS,DISP=(OLD,DELETE)
//$RSTRANS DD DSN=$CAI99.BILL.JUNE10,DISP=(,CATLG),
//            UNIT=SYSDA,SPACE=(TRK,(10,5))
//$RSEXT2I DD DSN=$CAI01.$RS.HIST(0),DISP=SHR
//$RSEXT2O DD DSN=$CAI01.$RS.HIST(+1),DISP=(,CATLG),
//            UNIT=SYSDA,SPACE=(TRK,(2,1))
//$RSEXT1I DD *
&U KBDAYS
&R 0.00018 3390 SYS
&R 0.00012 3390
&R 0.00024
//SYSIN    DD *
$RSVP INBILL(BILL) LIST(CHARGES) BLIST(BILLLIST) -
HEADING1('1          ACCOUNT HISTORY, JUNE - 1998') -
NOPRINT SUBTOT SORT(ACCOUNT)
```

First Step

Notice the control statements in $RSEXT1I on each step. On the first step, they specify billing units (&U) of KBDAYS. The units specified must be in the billing list pointed to by BLIST to get any charges. The rates (&R) must be in order of most specific to most general. The first rate is 0.00018 dollars per kilobyte-day (KBDAYS) for the 3390 volumes beginning with SYS. The second rate of 0.00012 dollars/KBDAYs applies to all other 3390 volumes. The last rate of 0.00024 dollars/KBDAY applies to all non-3390 volumes.

Besides the $RSEXT1I control statements, a registration file must be specified by $RSEXT2R on the first step. If a data set cannot be matched to an entry in the registration file, RSVP sets ACCOUNT to UNKNOWN. An additional DD statement

needed in the first step is $RSVOLTM which is a volume timestamp file. This file contains records indicating the last time billing was done on each volume. The billing detail records in the first step are going to a temporary $RSTRANS since they are modified by an ACCOUNT factor in the second step.

In the first step, the $RSVP command itself is simple. The LIST(CHARGDET) displays the ACCOUNT and current charges (CHCUR) on data sets. Any list would be sufficient or you could specify NOPRINT. The billing field list (BLIST) would default to BILLLIST but is specified for documentary reasons. A report title (BILLING DETAILS, JUNE - 1998) is specified in the HEADING1 keyword.

Second Step

In the second step, account rate factors are applied. Maintaining data in production libraries is encouraged, so account PROD gets a 20 percent discount (that is, a rate factor of 0.80 is applied to all records of account PROD). MARGE is the secretarial/clerical account and gets a rate factor of 0.95 or a 5 percent discount. The new account LEE began shortly after the last billing run and gets an additional charge of 20 percent to make up for the extra days by using an account factor of 1.20.

In the second step, $RSEXT2R and $RSVOLTM are not needed, but the billing history is specified with the DD statements $RSEXT2I (prior billing history) and $RSEXT2O (new history). Since ACCOUNT factors are applied in the second step, the $RSTRANS (billing detail records) DD statement specifies the actual billing information.

The most important keyword in the second step is INBILL, which defines the DDNAME of the billing detail records from the first step. Again BLIST is documentary since BILLLIST is the default. INBLIST is left out since it defaults to whatever is specified for BLIST. To accumulate the billing history properly, it is necessary to specify SORT(ACCOUNT) so the detail records are in the proper order. The keywords NOPRINT and SUBTOTAL print a summary of charges by ACCOUNT. The LIST(CHARGES) produce not only current charges (CHCUR) but month-to-date (CHMTD) and year-to-date (CHYTD) charges as well.

The output from this JCL is shown on the following pages.

This is the report generated from the first step:

```
          BILLING DETAILS, JUNE - 1998             PAGE  1

ACCOUNT KBALLOC KBDAYS CHCUR UNIT VOLUME DSNAME

DICK     1983   59490   7.14 3390 USER02 $CAI00.AJ.CNTL
DICK     1144   34320   4.12 3390 WORK01 $CAI00.CDG.ASM
DICK     1144   34320   4.12 3390 WORK02 $CAI00.CDG.ASM
DICK     4958  148740  17.85 3390 USER01 $CAI00.STEPLIB.LOAD
TOM      1430   24310   4.38 3390 SYS500 $CAI01.$RS.ASM
TOM       572   17160   3.09 3390 SYS500 $CAI01.$RS.MACS.ASM
TOM      7437  223110  40.16 3390 SYS500 $CAI01.$RS.PRINTAA
TOM        95       0   0.00 3390 USER01 $CAI01.$RS.TRAN
MARGE     572   17160   3.09 3390 SYS500 $CAI02.EXPECT.DATA
MARGE    1716   51480   6.18 3390 USER01 $CAI02.LABELS.DATA
MARGE     839   25170   3.02 3390 USER02 $CAI02.TTM.SOURCE
MARGE     572   17160   2.06 3390 USER02 $CAI02.TTM.SRCBKUP
JOE       572   17160   3.09 3390 SYS500 $CAI03.ARCH.DATA
JOE       763   22890   2.75 3390 USER01 $CAI03.AUTH.LOAD
JOE       953   28590   3.43 3390 USER02 $CAI03.DG.ASM
JOE       973   24325   2.92 3390 WORK02 $CAI03.LOAD
LEE      2021   60630   7.28 3390 USER01 $CAI04.MIGRATE.ASM
LEE      5873  176190  21.14 3390 USER01 $CAI04.MIGRATE.LIST
LEE      1239   37170   4.46 3390 USER02 $CAI04.MIGRATE.OBJ
LEE      1716   24024   4.32 3390 SYS500 $CAI05.ASM2MAN.TEXT
LEE       572   17160   3.09 3390 SYS500 $CAI05.CMD.CLIST
LEE      1144   34320   6.18 3390 SYS500 $CAI05.MACLIB.ASM
LEE      5721  171630  30.89 3390 SYS500 $CAI05.SOURCE.PAN
UNKNOWN   191     955   0.11 3390 USER02 $CAI06.MISC.CNTL
UNKNOWN   191     955   0.11 3390 USER02 $CAI06.MISC.TEXT
UNKNOWN   191     191   0.02 3390 USER02 $CAI06.SPFLOG2.LIST
PROD      305    9150   1.10 3390 USER01 $CAI98.L703.DATA
PROD      305    9150   1.10 3390 USER01 $CAI98.T703.CARDS
PROD     4405  132150  23.79 3390 SYS500 $CAI99.$RSSRC.LIST
PROD     4004  120120  21.62 3390 SYS500 $CAI99.CALLSEQ.DATA
PROD     5206  156180  28.11 3390 SYS500 $CAI99.SOURCE.ASM
PROD    11441  343230  61.78 3390 SYS500 $CAI99.STEPLIB.LIST
PROD     2288   68640  12.36 3390 SYS500 $CAI99.STEPLIB.LOAD

        82313 4999007 796.93            TOTAL
```

The second step shows the ACCOUNT summaries:

```
          ACCOUNT HISTORY, JUNE 10, 1998          PAGE    1

ACCOUNT  DSCOUNT  KBALLOC  KBDAYS    CHCUR    CHMTD    CHYTD

LEE           18       0   292253    62.98    62.98    62.98
MARGE         70       0   256647    30.55    30.55  2546.44
DICK          34       0   395011    47.44    47.44  4993.73
LEE           27       0   372084    44.65    44.65  1610.67
PROD          91       0  2717349   384.79   384.79 29270.49
TOM           21       0   333314    56.07    56.07  3615.32
JOE           70       0   630248    83.13    83.13  3154.92
UNKNOWN        3       0     2101     0.24     0.24  1174.06

TOTAL        334       0  4999007   709.85   709.85 46428.61
```

The billing detail records from $RSTRANS for the first step follow:

```
07/03/00 3390  USER02 $CAI00.AJ.CNTL         DICK     1983   59490   7.14
07/03/00 3390  WORK01 $CAI00.CDG.ASM         DICK     1144   34320   4.13
07/03/00 3390  WORK02 $CAI00.CDG.ASM         DICK     1144   34320   4.13
07/03/00 3390  USER01 $CAI00.STEPLIB.LOAD    DICK     4958 1487401   7.85
07/03/00 3390  SYS500 $CAI01.$RS.ASM          TOM     1430   24310   4.38
07/03/00 3390  SYS500 $CAI01.$RS.MACS.ASM     TOM      572   17160   3.09
07/03/00 3390  SYS500 $CAI01.$RS.PRINTAA      TOM     7437  223110  40.16
07/03/00 3390  SYSER0 $CAI01.$RS.TRAN         TOM       95       0   0.00
07/03/00 3390  SYS500 $CAI02.EXPECT.DATA    MARGE     572   17160   3.09
07/03/00 3390  USER01 $CAI02.LABELS.DATA    MARGE    1716   51480   6.18
07/03/00 3390  USER02 $CAI02.TTM.SOURCE     MARGE     839   25170   3.02
07/03/00 3390  USER02 $CAI02.TTM.SRCBKUP    MARGE     572   17160   2.06
07/03/00 3390  SYS500 $CAI03.ARCH.DATA        JOE     572   17160  3.09
07/03/00 3390  USER01 $CAI03.AUTH.LOAD        JOE     763   22890  2.75
07/03/00 3390  USER02 $CAI03.DG.ASM           JOE     953   28590  3.43
07/03/00 3390  WORK02 $CAI03.LOAD            JOE     973   24325  2.92
07/03/00 3390  USER01 $CAI04.MIGRATE.ASM      LEE    2021   60630   7.28
07/03/00 3390  USER01 $CAI04.MIGRATE.LIST     LEE   58731  761902   1.14
07/03/00 3390  USER02 $CAI04.MIGRATE.OBJ      LEE    1239   37170   4.46
07/03/00 3390  SYS500 $CAI05.ASM2MAN.TEXT     LEE    1716   24024   4.32
07/03/00 3390  SYS500 $CAI05.CMD.CLIST        LEE     572   17160   3.09
07/03/00 3390  SYS500 $CAI05.MACLIB.ASM       LEE    1144   34320   6.18
07/03/00 3390  SYS500 $CAI05.SOURCE.PAN       LEE    5721  171630  30.89
07/03/00 3390  USER02 $CAI06.MISC.CNTL    UNKNOWN     191     955   0.11
07/03/00 3390  USER02 $CAI06.MISC.TEXT    UNKNOWN     191     955   0.11
07/03/00 3390  USER02 $CAI06.SPFLOG2.LIST UNKNOWN     191     191   0.02
07/03/00 3390  USER01 $CAI98.L703.DATA       PROD     305    9150   1.10
```

And the result of the second step follows:

```
07/03/00 3390  SYS500 $CAI05.SOURCE.PAN       LEE    57211  716303  7.07
07/03/00 3390  SYS500 $CAI05.MACLIB.ASM       LEE     1144   34320  7.42
07/03/00 3390  SYS500 $CAI05.CMD.CLIST        LEE      572   17160  3.71
07/03/00 3390  SYS500 $CAI05.ASM2MAN.TEXT     LEE     1716   24024  5.18
07/03/00 3390  USER02 $CAI02.TTM.SOURCE     MARGE      839   25170  2.87
07/03/00 3390  USER02 $CAI02.TTM.SRCBKUP    MARGE      572   17160  1.96
07/03/00 3390  USER01 $CAI02.LABELS.DATA    MARGE     1716   51480  5.87
07/03/00 3390  SYS500 $CAI02.EXPECT.DATA    MARGE      572   17160  2.94
07/03/00 3390  USER01 $CAI00.STEPLIB.LOAD    DICK    49581  487401  7.85
07/03/00 3390  WORK02 $CAI00.CDG.ASM         DICK     1144   34320  4.12
07/03/00 3390  WORK01 $CAI00.CDG.ASM         DICK     1144   34320  4.12
07/03/00 3390  USER02 $CAI00.AJ.CNTL         DICK     1983   59490  7.14
07/03/00 3390  USER02 $CAI04.MIGRATE.OBJ      LEE     1239   37170  4.46
07/03/00 3390  USER01 $CAI04.MIGRATE.LIST     LEE    58731  761902  1.14
07/03/00 3390  USER01 $CAI04.MIGRATE.ASM      LEE     2021   60630  7.28
07/03/00 3390  USER01 $CAI98.L703.DATA       PROD      305    9150  0.88
07/03/00 3390  USER01 $CAI01.$RS.TRAN         TOM       95       0  0.00
07/03/00 3390  SYS500 $CAI01.$RS.PRINTAA      TOM    74372  231104  0.16
07/03/00 3390  SYS500 $CAI01.$RS.MACS.ASM     TOM      572   17160  3.09
07/03/00 3390  SYS500 $CAI01.$RS.ASM          TOM     1430   24310  4.38
07/03/00 3390  WORK02 $CAI03.LOAD            JOE      973   24325  2.92
07/03/00 3390  USER02 $CAI03.DG.ASM           JOE      953   28590  3.43
07/03/00 3390  USER01 $CAI03.AUTH.LOAD        JOE      763   22890  2.75
07/03/00 3390  SYS500 $CAI03.ARCH.DATA        JOE      572   17160  3.09
07/03/00 3390  USER02 $CAI06.SPFLOG2.LIST UNKNOWN      191     191  0.02
07/03/00 3390  USER02 $CAI06.MISC.TEXT    UNKNOWN      191     955  0.11
07/03/00 3390  USER02 $CAI06.MISC.CNTL    UNKNOWN      191     955  0.11
```

Notice the change in dollar amounts for the various accounts that had account factors in the second step.

The history input $RSEXT2I looks like this:

```
  Date       YTD      MTD       Jan   |    Apr    May   |  Account

05/29/00   2515.89   476.97   487.35---- 539.21   0.00----MARGE
03/29/00   1423.09   512.13   423.64----   0.00   0.00----DAVE
05/29/00   4946.33   998.98   959.12---- 993.47   0.00----DICK
05/29/00   1566.02   314.36   294.78---- 319.79   0.00----LEE
05/29/00  28885.71  5843.69  5667.32----5843.69   0.00----PROD
05/29/00   3559.25  1017.50     0.00----1017.50   0.00----TOM
05/29/00   3071.79   687.21   327.86---- 695.67   0.00----JOE
04/29/00    436.19   436.19     0.00----   0.00   0.00----TESTDATA
02/29/00   1173.82   586.91   586.91----   0.00   0.00----UNKNOWN
```

And the history output follows:

```
  Date       YTD      MTD      Jan   |    Apr     May   |  Account

07/03/00     62.98    62.98     0.00----   0.00    0.00----LEE
07/03/00   2546.44    30.55   487.35---- 539.21  476.97----MARGE
03/29/00   1423.09   512.13   423.64----   0.00    0.00----JOANNA
07/03/00   4993.73    47.44   959.12---- 993.47  998.98----DICK
07/03/00   1610.67    44.65   294.78---- 319.79  314.36----LEE
07/03/00  29270.49   384.79  5667.32----5843.69 5843.69----PROD
07/03/00   3615.32    56.07     0.00----1017.50 1017.50----TOM
07/03/00   3154.92    83.13   327.86---- 695.67  687.21----JOE
04/29/00    436.19   436.19     0.00----   0.00    0.00----TESTDATA
07/03/00   1174.06     0.24   586.91----   0.00    0.00----UNKNOWN
```

The registration file for this run follows:

```
L$CAI00                         DICK
L$CAI01                         TOM
L$CAI02                         MARGE
L$CAI03                         JOE
L$CAI04                         LEE
L$CAI05                         LEE
L$CAI98                         PROD
L$CAI99                         PROD
```

The registration file can include more information. You can enter up to the full 44-character data set name, and also specify a unit type and volume. When multiple entries are present that start with the same data set name, they must be in order from most specific to most general.

4.3 One-Step Billing

There are four methods of determining account information for One-Step billing. The primary sort field must be ACCOUNT to maintain proper history information. The four methods are:

1. Use the CA-ASM2 OPEN modifications to IFG0196W to put the account number instead of the last modified job name in the Format-1 DSCB. Update the user CSECT (see 5-9 in the "Tailoring" chapter) to pick up this account field. The necessary update is in comments in the user CSECT. (Check $CNVAGE for any change in displacement.)
2. Use a user modification to the open SVC that saves the account number in the Format-1 DSCB. You must modify the user CSECT to get the account number from the correct position in the DSCB after $CNVAGE has rearranged the DSCB.
3. Modify $RSEXT1 (see RSVP User Exits on page 5-4 in the "Tailoring" chapter) to fill in the account number according to whatever scheme may be valid in your installation.
4. Write a format exit to place the account number into the formatted DSCB. Modify the user CSECT to obtain the account number from a format exit.

4.3.1 Sample JCL for One-Step Billing

```
//BILL     EXEC ASM2CMDU
//$RSOUT   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=A
//$RSTRANS DD DSN=$CAI99.BILL.JUNE10,DISP=(,CATLG),
//            UNIT=SYSDA,SPACE=(TRK,(10,5))
//$RSEXT2I DD DSN=$CAI01.$RS.HIST(0),DISP=SHR
//$RSEXT2O DD DSN=$CAI01.$RS.HIST(+1),DISP=(,CATLG),
//            UNIT=SYSDA,SPACE=(TRK,(2,1))
//$RSVOLTM DD DSN=$CAI01.$RS.VOLTM,DISP=OLD
//$RSEXT1I DD *
&U KBDAYS
&R 0.00018 3390 SYS
&R 0.00012 3390
&R 0.00024
&A 1.20 LEE
&A 0.95 MARGE
&A 0.80 PROD
//SYSIN    DD *
$RSVP VOL(ALL) LIST(CHARGES) BLIST(BILLLIST)
HEADING1('1           ACCOUNT HISTORY, JUNE - 1998')
NOPRINT SUBTOT SORT(ACCOUNT)
```

Chapter 5. Tailoring

User exits and a user CSECT are provided for modifying and customizing RSVP to the specific requirements of your environment.

The user CSECT contains system parameters, field definitions, and output lists used by RSVP. Special macros are distributed in the base MAC library that you can use to add, change, or replace any of the specifications contained in the user CSECT.

Note: Users of CA-1 and CA-TLMS may need to reassemble $RSUSER based on their release level. CAIMAC member $TAPEFMT contains the variable &EARLFMT which is dependent on the level of tape management system. The default for &EARLFMT is OLD. If you are CA-1 Version 5.2 or higher, or TLMS Version 5.5 or higher, you need to change the value. See CAIMAC member $TAPEFMT for the appropriate values.

5.1 JCL

Besides the minimum JCL needed for each operating system, additional DD statements may be desirable or necessary for the operation of RSVP. You can change the attributes and DDNAMEs of all DD statements used in either the user CSECT (see 5-9) or one of the exits (see 5-4).

DDNAME	Description
$RSOUT	Report output. If this DD statement is not present, TSO TPUTs are used for the report. This directs the output to SYSTSPRT but eliminates heading lines and page breaks since a terminal does not use print controls. The default attributes for $RSOUT are: ■ Record format FBA (Fixed-Block-with ANSI control characters). ■ Record length 133 (DEFLEN in user CSECT, also changed by CHARS keyword). ■ Block size 1330 (BLSYSOUT in user CSECT, overridden by CHARS if specified). ■ DDNAME $RSOUT (DDSYSOUT in user CSECT, also changed by DDNAME keyword).
$RSTRANS	Transaction file, generally used for preprocessing and billing. The default values are: ■ Record format FB (Fixed-length Blocked). Use RFBILL in the user CSECT to change it to VB (Variable-length Blocked) and make it resemble an SMF record. ■ Record length 132 (LRBILL in user CSECT). ■ Block size 6204 (BLBILL in user CSECT). ■ DDNAME $RSTRANS (DDBILL in user CSECT). In addition to the normal user CSECT, $RSUSER8 is set to give a record size of 80 and block size of 6160 to $RSTRANS to allow generating 80-byte card images for preprocessing of IBM utilities. This can be specified with the USERCSECT keyword. See User CSECT on page 5-9 for details. See member $RSUSER8 in the BASE SRC library for an example on how to change the transaction file default values with the $RSUD macro.

$RSVOLTM Volume timestamp file used for billing. All fields are in character format to allow editing, if necessary. The defaults are:

- Record format FB (Fixed-length Blocked).
- Record length 20 (LRBILLTM in the user CSECT).
- Block size 6120 (BLBILLTM in the user CSECT).
- DDNAME $RSVOLTM (DDBILLTM in the user CSECT).

See Macro Overview on page 5-10 for details on using the $RSUD macro to change the default values for the timestamp file. The record layout is defined by the DSECT macro $RSDIR:

Column	Format	Description
1-6	Chars	Volume Serial
7-11	Chars	Date last billed
12-15	Chars	Time last billed
16-20	Chars	Reserved for future use

5.2 RSVP User Exits

The following RSVP user exists are described in this section:

- $RSBILL
- $RSEXT1
- $RSEXT2
- $RSEXT10
- $RSEXT11
- $RSEXT12

5.2.1 $RSBILL ($RSVP Command Billing)

$RSBILL modifies the method by which the billing transaction record is written. Upon entry, register 1 contains the address of the common area ($RSCOM). EXITF1 has the EXITF1L bit set for the cleanup call. You may want to reduce the actual number of billing records written by doing some subtotaling.

$RSBILL is an integral part of RSVP and must be linked back into RSVP properly. The link-edit control cards are in LINKRSVP; ASM$RSVP contains the JCL to assemble the RSVP user exits. Since $RSBILL is linked into RSVP, you must ensure that your changes are proper for all functions that write to the transaction file ($RSTRANS), such as preprocessing.

5.2.2 $RSEXT1

The selection exit $RSEXT1 lets you include or exclude data sets based on an input file. $RSEXT1 receives control before a data set is finally selected, and has the option of rejecting the data set.

The main reasons for modifying $RSEXT1 are to:

- Include special selection criteria not already present.
- Change the DCB characteristics of the input file ($RSEXT1I).
- Insert the account number based on fields in the DSCB.
- Change the values associated with one of the fields.

You can change the name of $RSEXT1 in the user CSECT (see 5-9). With multiple user CSECTs, it is possible to invoke multiple exits.

5.2.2.1 Exit Router Facility

You can use the Exit Router facility to add new exit functions. It can route control to up to ten additional exit modules. This allows your installation additional exit function or data set selection criteria without modifying $RSEXT1 and without having to code a user CSECT in some cases.

To specify additional exit modules to be called by $RSEXT1, code from one to ten //$RSEXT1I DD input control statements that start in column 1 and have the form:

```
&Xn
```

where n is any alphabetic or numeric character.

The name of the exit module called by $RSEXT1 is $RSEXT1n, where

```
n=0,1,2,...,9 are reserved for RSVP development
n=A,B,C,...,Z are available for user exit modules
```

The $RSEXT1n exit modules are given control after both RSVP and the $RSEXT1 exit have selected a data set for processing or reporting. The $RSEXT1n exit modules can reject data sets by additional criteria or modify selected data fields.

A sample exit module, $RSEXT10, demonstrates the exit router facility. $RSEXT10 writes a message when it is called at the initial and termination calls from $RSEXT1. An example of a $RSVP command using the $RSEXT10 sample exit is shown here:

```
//$RSVP    EXEC RSVPCMDU
//SYSIN    DD   *
$RSVP IF(DSORG EQ DA) PRINT(NEW (DSNAME USED)) -
VOLUME(123456)
*
//$RSEXT1I   DD   *
&X0  /*INVOKE $RSEXT10 SAMPLE EXIT */
```

The output from this example lists all DSORG=DA data sets on pack 123456, with a message from module $RSEXT10 before the list of data sets, and another message from module $RSEXT10 after the list of data sets.

Input Stream

To invoke the router facility, include the following in the RSVP input stream:

```
//$RSEXT1I DD *
&X1             Identify empty data sets
&X2             Exclude not empty data sets
/*
```

where:

$RSEXT1I Is input to EXIT1, the Selection Exit. The defaults are: record format FB, record length and block size assumed from the JCL, but the buffer is a maximum of 80 bytes. The file definition can be changed, if desired.

Control Statements

Parameter Description

&X1 Causes the $RSEXT11 exit to be called. $RSEXT11 sets the TYPE field to the value EMTY for each DSORG as shown next. You must specify the TYPE and USED fields explicitly or implicitly in the $RSVP command. They can be in a LIST, PRINT, NOPRINT, SORT, BLIST, INBLIST, or AUXFLDS list.

DSORG Description

PS/BDAM/VSAM/unknown
Sets TYPE=EMTY if USED=0. For VSAM data sets, you must specify both the VOLUME and CLUSTER parameters.

ISAM Obtains the Format-2 DSCB and sets TYPE=EMTY if the Format-2 DSCB shows the data set has never been loaded (bit X'20' not set in DS2STIND).

PO Calls $SM to open the data set, read the first directory block, close the data set, and if no member is defined, sets TYPE=EMTY. Note the additional overhead required for this.

&X2 This must follow &X1. It causes the $RSEXT12 to be called, and excludes data sets that do not have the TYPE field set to EMTY.

Note: The TYPE field cannot be used in an IF clause, since its value is set by the $RSEXT1 and $RSEXT11 exits after RSVP performs the selection based on IF clause criteria.

5.2.3 $RSEXT2

$RSEXT2, the Post-Sort exit, is called just before printing. It lets you modify the output after the sort. $RSEXT2 can set a return code of 4 in register 15 to cause a record to be deleted from further processing. It can insert records by passing a return code of 12. After the current record is processed, $RSEXT2 receives control to build the next (inserted) record.

You can change the name of $RSEXT2 in the user CSECT (see 5-9). With multiple user CSECTs, it is possible to invoke multiple exits.

You may want to modify $RSEXT2 to change the various DCB attributes ($RSEXT2R, $RSEXT2I, $RSEXT2O), the method of obtaining the account number, or method of maintaining the account history.

$RSEXT2 contains these files:

File Name	Description
$RSEXT2R	Registration file used in EXIT2, to match data set names to an account. The defaults are: record format FB, record length and block size default to JCL, but the buffer size is 120. This file can be changed, if desired. See $RSEXT2R on page 4-8 in the "DASD Billing" chapter for the record format and additional information.
$RSEXT2I	Account Billing History file used in EXIT2. The defaults are record format FB, record length and block size assumed from the JCL (but it is assumed to be a previous output history $RSEXT2O that is record length 150, block size 6000), but the buffer length is 150. This file can be changed, if desired. It is input from a prior run and the current charges are added to it to produce a new history going to $RSEXT2O. See $RSEXT2I on page 4-9 in the "DASD Billing" chapter for additional information.
$RSEXT2O	History output file generated by EXIT2. The defaults are: record format FB, record length 150, and block size 6000. The file can be changed, if desired. $RSEXT2O contains the input from $RSEXT2I with current charges added to it. See $RSEXT2O on page 4-9 in the "DASD Billing" chapter for additional information.

5.2.4 $RSEXT10 (RSVP Sample &XO Control Statement Processor)

This exit is available for users who wish to code their own exit routines which get called from $RSEXT1. If you code &X1 in the $RSEXT1 DD control file, exit $RSEXT11 is invoked. $RSEXT10 could then be used as a skeleton for your own module, for example $RSEXT1B, and then you could code &XB in the $RSEXT11 DD input file. If you code &XO in the $RSEXT11 input file, the $RSEXT10 routine writes a message the first time it is entered and then simply returns to $RSEXT1 every other time it is called.

5.2.5 $RSEXT11 (RSVP Empty Data Set Identification Exit)

This exit sets the TYPE data item = EMTY for each DSORG as described next. It is given control if there is an &X1 control statement in the $RSEXT1I input file.

DSORG	Description
PS/BDAM/VSAM/unknown	Sets TYPE=EMTY if RSVP has set USED=0. For VSAM data sets the RSVP VOLUME and CLUSTER parameters must both be specified.
ISAM	Obtains the Format-2 DSCB and sets TYPE=EMTY if the Format-2 DSCB shows the data set has never been loaded (bit X'20' not set in DS2STIND).
PO	Calls $SM to open the data set, read the first directory block, close the data set, and if no member is defined sets TYPE=EMTY. Note the additional overhead required for this.

5.2.6 $RSEXT12 (RSVP Non-Empty Data Set Exclusion Exit)

This exit excludes data sets which are not TYPE=EMTY. It is given control if there is an &X2 control statement in the $RSEXT1I input file.

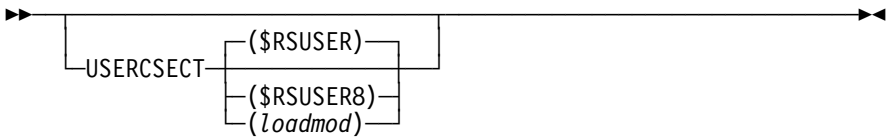
5.3 User CSECT

The RSVP user CSECT contains all field, report lists, and transaction definitions used by RSVP. The default name of the user CSECT is $RSUSER. You can customize the user CSECT or define multiple user CSECTs to your installation's requirements. Five macros are provided that help you define the user CSECT. These macros are described in Macro Overview on page 5-10.

Note: CA-1 and TLMS users should review the CAIMAC member $TAPEFMT and ensure that the correct value for &EARLFMT is used. These values determine how $RSUSER and other RSVP USERCSECTS are assembled. You may need to reassemble $RSUSER to run RSVP with your tape system.

5.3.1 Command Format for User CSECT

Command Format for User CSECT



Keyword Descriptions

USERCSECT Specifies the load module name of the user CSECT for this execution of the command. For example: `USERCSECT(MYCSECT)`

The default is $RSUSER.

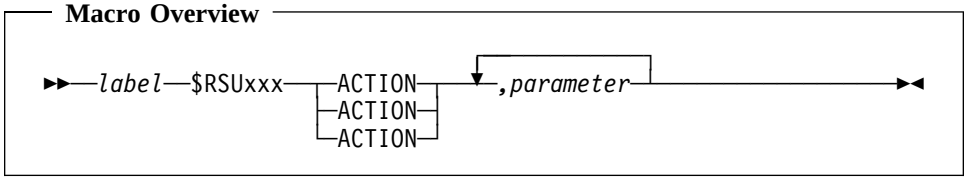
To allow generation of 80-byte card images to use as input to IBM utilities, an additional user CSECT exists ($RSUSER8) that has a record length of 80 for $RSTRANS.

5.4 Macro Overview

The five primary macros that define the user CSECT are written specifically for the IBM assembler. Therefore, the descriptions assume familiarity with IBM assembler conventions and the $RSVP command.

Macros	Description
$RSUD	Defines or modifies defaults. (For example, what value space to report on, how to report VSAM, and so on.) The two formats for the $RSUD macro are: ■ Defining file default values. ■ Defining other default values not related to files.
$RSUF	Defines or modifies fields that you can reference in the $RSVP command for listing, sorting, and so on.
$RSUL	Defines lists of fields used for printing and building transaction records.
$RSUV	Defines how to process VSAM dictionary entries. Entries are returned from a catalog locate request.
$RSUEND	Generates the user CSECT based on values set by the previous macro commands.

Syntax



Syntax Description

label Generates an assembler symbolic label used to reference the values defined on the macro command. It is required for $RSUF and $RSUV macros, optional for $RSUL macros and ignored on $RSUD and $RSUEND macros.

ACTION Defines the given parameters as modifications, replacements, or new additions. Valid actions are ADD, CHG, and REP. Not used on the $RSUEND macro.

parameter Specifies the values to be set for the macro. See the following sections for details of the individual parameters on each macro.

5.4.1 Hierarchical Use of the Five Macros

When assembling the user CSECT, there is a hierarchy in using these macros to control what user CSECTs they affect. A sample of $RSUSER8, the standard user CSECT (except for its generation of 80-byte records for IBM utilities) is shown here.

```
$RSUSER8 TITLE '$RSVP USER CSECT 80 BYTE TRANS RECORD'
         COPY  $RSASGBL     COPY GENERAL ASSEMBLY GLOBALS
         COPY  $RSASSET     COPY GENERAL ASSEMBLY SETC'S
         COPY  $RSUPRED     COPY PREDEFINED FIELDS, ETC
         COPY  $RSUSR00     COPY USER DEFINED FIELDS, ETC.
*
*     OVERRIDE LRECL, BLKSIZE, RECFM, SMFID ON $RSTRANS FILE
*     TO GENERATE 80 BYTE CARD IMAGES FOR IBM UTILITIES.
*
      $RSUD CHG,FILE=TRANS,LRECL=80,BLKSIZE=6160,RECFM=FB,           X
               SMFID=0
      $RSUEND
         END
```

The first two COPY statements for $RSASGBL and $RSASSET control the length and content of the assembly listing (if you used PRINT GEN for the entire assembly, the listing would be several thousand lines long). The statement COPY $RSUPRED is the first to include the macros being discussed.

$RSUPRED is the member of the base MAC library having the defaults, fields, and lists defined in RSVP. It can be modified at any time. However, Computer Associates recommends that you not make any modifications to this member unless instructed to do so. By not changing it, you do not have to refit your modifications every time $RSUPRED is replaced.

The next statement, COPY $RSUSR00, copies a second member of the BASE MAC library, $RSUSR00, which is reserved for your standard modifications (those that should be made to every copy of the user CSECT generated at your installation).

This member ($RSUSR00) is also used to resolve field names in all exits. Therefore, if you define a new field that is to be addressable by an exit (using the $RSF macro, not defined here), it must be defined in this member. Since changes are not made to this member between versions, you do not have to refit modifications defined in $RSUSR00. At most, you just have to reassemble your user CSECTs.

Note: If a specific group of modifications apply to multiple user CSECTs, but not to all user CSECTs, code members $RSUSR01 to $RSUSR99 with the modifications and add additional COPY statements for the member(s) needed.

In the example, an additional $RSUD macro follows the last COPY statement. You may code any number of $RSUD, $RSUF, $RSUL, and $RSUV macros. These macros apply ONLY to this version of the user CSECT.

The last macro statement of any user CSECT must be the $RSUEND macro to generate the actual user CSECT.

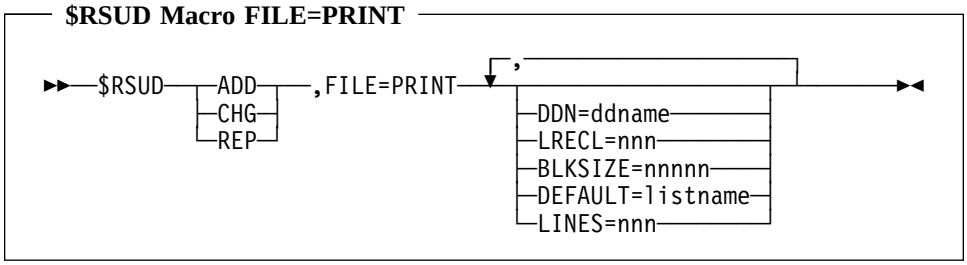
5.4.2 $RSUD Macro

$RSUD: Defining FILE Default Values: Shown next are three cases using the macro with the FILE parameter. In the first case, FILE equals PRINT; in the second, TRANS; in the third, TIMESTAMP.

Review the syntax cases and the conventions before coding RSVP macros. Be aware of these two points when using $RSUD macros:

- "label" is ignored with the $RSUD macro.
- BLKSIZE and LRECL parameters follow the standard IBM conventions.

Syntax 1: FILE=PRINT

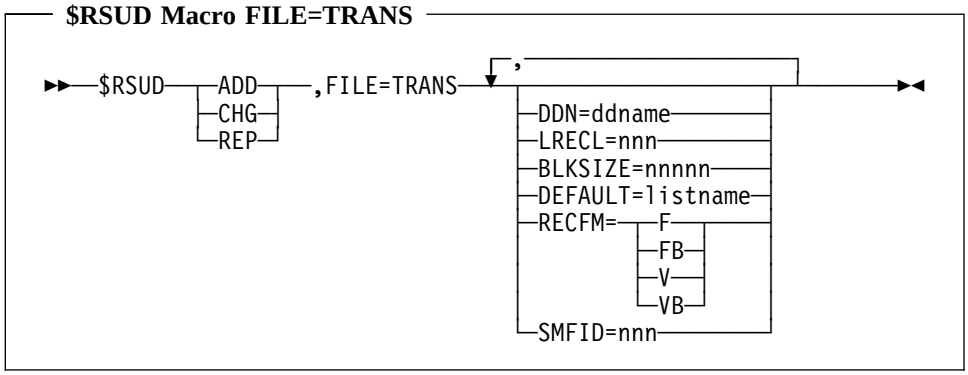


Syntax Description:

When using the FILE=PRINT parameter, an action of ADD or REP must specify the parameters DDN=ddname, LRECL=nnn, and BLKSIZE=nnnnn.

If the CHG action is used, at least one of the parameters is required.

Syntax 2: FILE=TRANS

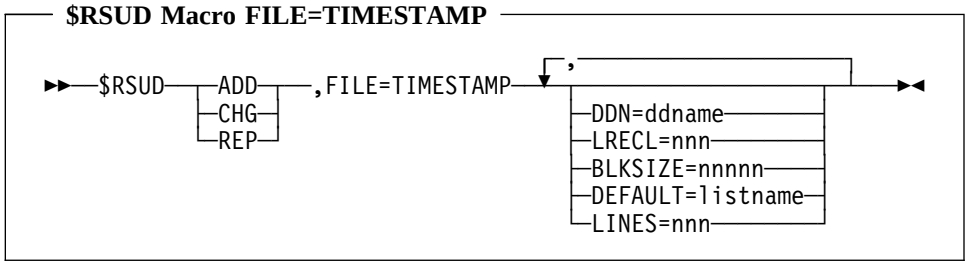


Syntax Description:

When specifying an action of ADD or REP with FILE=TRANS, you must specify the parameters DDN=ddname, LRECL=nnn, BLKSIZE=nnnnn, and RECFM=xx.

If the CHG action is used, at least one of the parameters is required.

Syntax 3: FILE=TIMESTAMP



Syntax Description:

When specifying an action of ADD or REP with FILE=TIMESTAMP, you must specify all three parameters:

`DDN=ddname, LRECL=nnn, BLKSIZE=nnnnn`

Actions:

ADD The file indicated by FILE has not previously been defined. The definition on this $RSUD macro command is complete for this file and includes all required parameters.

CHG The file indicated by FILE has been previously defined. The parameters specified on this $RSUD macro command override the parameters specified on previous related macros.

REP The file indicated by FILE has been previously defined. However, the previous definition is to be totally disregarded. All required parameters are reentered on this $RSUD macro command. Any optional parameters are reset to the default.

Parameters:

FILE

Identifies the file to which the remaining parameters apply. The valid operands of this keyword are:

TRANS	Transaction File
PRINT	Print (list) File
TIMESTAMP	Volume Timestamp File (used for billing)

DDN

DDNAME to be assigned to the file indicated. It must be a valid DDNAME (1-8 alphanumeric characters beginning with an alphabetic or special character).

LRECL

Logical record length to be assigned to the named file. The maximum record length is 256.

BLKSIZE

Physical block size to be assigned to the indicated file. It must be a multiple of LRECL with the exception that if RECFM=VB or V, on the TRANS file, then BLKSIZE must be at least four greater than LRECL.

DEFAULT

Identifies the list definition (defined by $RSUL macro) that is used to identify the fields to be included on the indicated file if no other list is defined by the actual $RSVP command.

LINES

Number of lines per page, including heading lines for the print file.

RECFM

Specifies the record format for the transaction file. Valid formats include:

F Fixed-length unblocked
FB Fixed-length blocked
V Variable-length unblocked
VB Variable-length blocked (if you wish to make records resembling SMF records)

SMFID

If the transaction file is built to resemble SMF records, this is the SMF record identifier to use. Valid values are 0 (not SMF records) or 200-255 (valid SMF user record identifiers). If a value other than 0 is specified, every transaction record has an SMF prefix preceding the fields selected:

SMFFLAG	X	Operating system flag
SMFRECID	X	SMF record ID (SMFID)
SMFTIME	XL4	Time (binary) for SMF timestamp
SMFDATE	XL4	Date (binary) for SMF timestamp
SMFSID	CL4	System ID from SMCA

5.4.2.1 Example 1 - FILE=TRANS

The statement `$RSUD REP,FILE=TRANS,LRECL=200,BLKSIZE=6200,RECFM=FB` causes a transaction file to be generated with 200-byte records and 31 records per block.

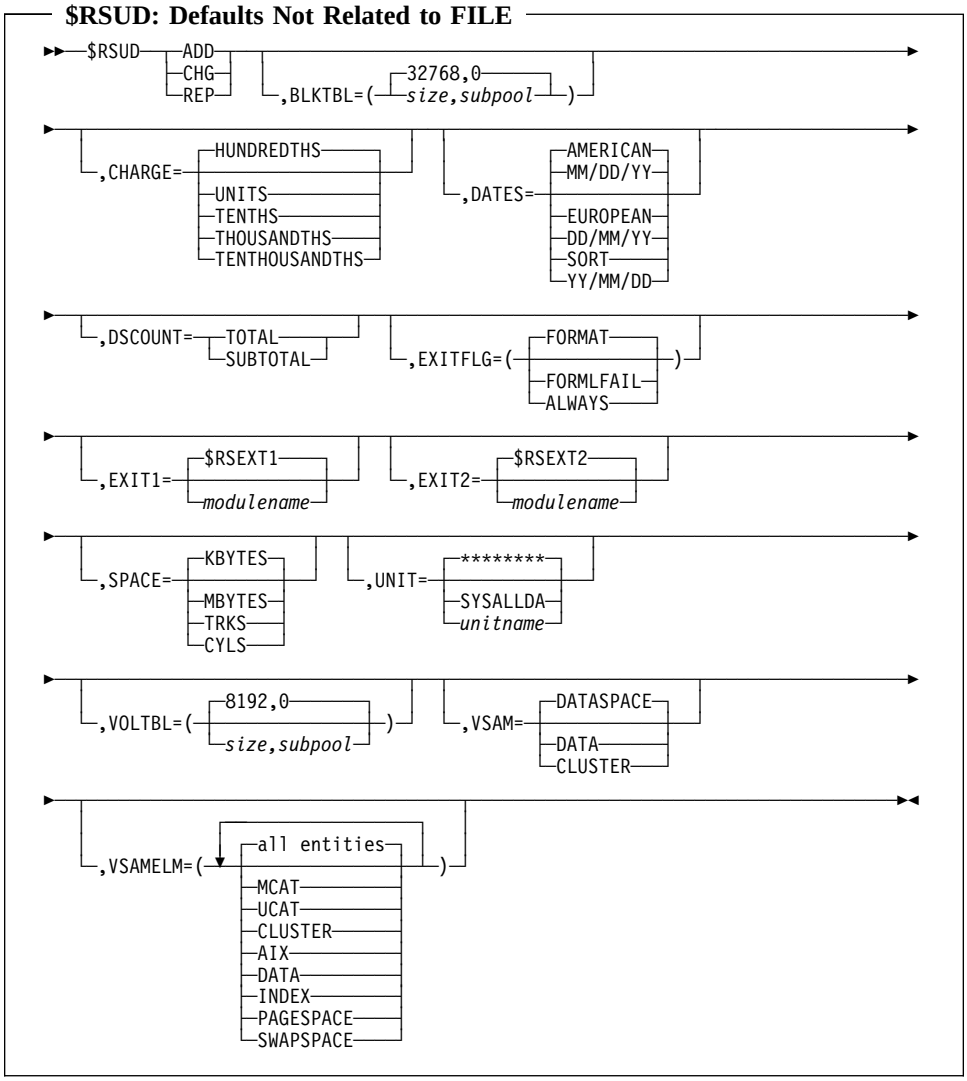
5.4.2.2 Example 2 - FILE=PRINT

The statement `$RSUD CHG,FILE=PRINT,LINES=75` causes the print file ($RSOUT DD) to print 75 lines per page.

5.4.3 $RSUD: Defaults Not Related to FILE

The syntax for specifying default values other than FILE defaults follows:

Syntax



Syntax Description:

When using other than FILE defaults with an action of ADD, CHG, or REP, you must use at least one of the parameters.

Actions:

ADD The parameters on this $RSUD macro command have not been previously specified.

CHG The parameters on this $RSUD macro command have been previously specified and the values listed here are the only parameters changed from the previous definitions.

REP The parameters on this $RSUD macro command have been previously defined but are to be overridden by this macro command. Parameters not entered are reset to the defaults.

Parameters:

BLKTBL

Values to be used for GETMAINs for the internal data set blocks (holding internal fields after formatting). The default is BLKTBL=(32768,0).

CHARGE

Defines the number of decimal places to be reserved in the charge fields (CHCUR, CHMTD, CHYTD, CHMONTHS). Useful only for DASD billing. Specifying UNITS indicates no decimal places are to be reserved. The default is hundredths.

DATES

Defines the format of all date fields (having FLAG=DATE on $RSUF macro) in the transaction and print files. The possible formats are:

AMERICAN or MM/DD/YY
EUROPEAN or DD/MM/YY
SORT or YY/MM/DD

The default is AMERICAN.

DSCOUNT

Defines when the value in the DSCOUNT field is printed. Specifying TOTAL causes it to be a running total (that is, the position of the data set in the list). Specifying SUBTOTAL causes it to be a running subtotal (that is, the position of the data set since the last subtotal line). The default is to print value only on a subtotal and total line.

EXITFLG

A flag indicating when EXIT1 should be invoked. Valid options are:

FORMAT	After the internal data set block is formatted.
FORMFAIL	Call if formatting failed for the data set.
ALWAYS	Call EXIT1 regardless of the result of formatting.

The default is FORMAT.

EXIT1

Load module name to be used for the first exit point (data set selection, presort). The default is $RSEXT1.

EXIT2

Load module name to be used for the second exit point (postsort, prelist and pretransaction). The default is $RSEXT2.

SPACE

Space values (ALLOC, USED, UNUSED) are to be reported in the units indicated:

MBYTES	Megabytes
KBYTES	Kilobytes
TRKS	Tracks
CYLS	Cylinders

The default is KBYTES.

UNIT

Unit name to be used in dynamic allocation of the volumes to be searched. If a unit name of ******** or SYSALLDA is used, the unit name SYSALLDA is used. This is a unit name known internally to IBM logic that applies to all currently mounted DASD units. The default is ********.

VOLTBL

Values to be used in GETMAINs for internally formatted volume blocks. The default is VOLTBL=(8192,0).

VSAM

Defines the level at which VSAM entities are to be reported.

DATA	VSAM entities are to be reported at the individual (data and index) component level.
CLUSTER	VSAM entities are to be reported at the cluster level.
DATASPACE	VSAM entities are to be reported directly from the Format-1 DSCB as data spaces. This reduces command execution time since no search of the VSAM catalog is needed.

The default is DATASPACE.

VSAMELM

Defines the VSAM entities to be included when VSAM is selected at the cluster or data/index level:

MCAT	Master Catalog
UCAT	User Catalogs
CLUSTER	Clusters
AIX	Alternate Indexes
DATA	Data (when listing at data/index level)
INDEX	Index (when listing at data/index level)
PAGESPACE	MVS page spaces
SWAPSPACE	MVS swap spaces

The default is all entities.

5.4.3.1 Example 1 - $RSUD Using REP

The statement:

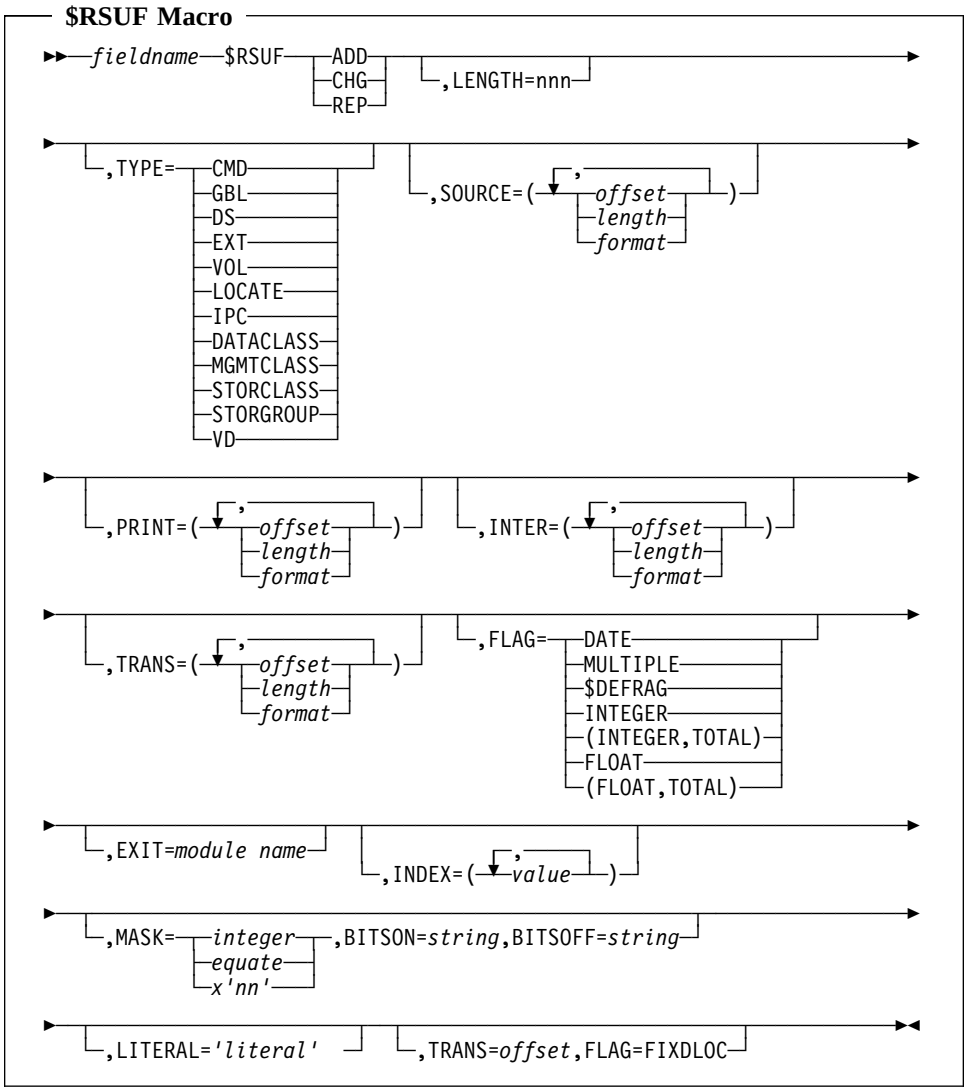
```
$RSUD REP,VSAMELM=(CLUSTER,DATA,INDEX),VSAM=CLUSTER
```

causes RSVP not to report on VSAM catalogs, alternate indexes, page spaces, and swap spaces. The statement also causes VSAM to be reported on at the cluster level unless this is overridden in the $RSVP command.

5.4.4 $RSUF Macro

This macro defines or modifies fields that you can reference in the $RSVP command for listing, sorting, and so on.

Syntax

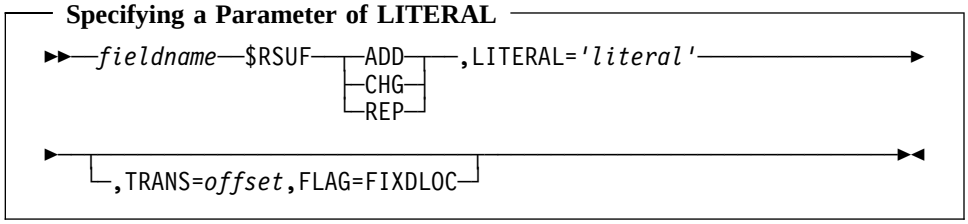


Syntax Description:

If you specify an action of ADD or REP, you must use the LENGTH=nnn parameter. You can also use any other parameters. Parameters not specified default. Defaults are TYPE=DS, offset=0, length=length specified by LENGTH=nnn above, format=character, and FLAG=no flags.

In all cases, fieldname is required. When specifying FLAG=DATE, you cannot specify other FLAG operands. When specifying the CHG action, you must use at least one parameter.

Syntax - Specifying a Parameter of LITERAL



Syntax Description:

TRANS=offset and FLAG=FIXDLOC are **optional**, but if you use either one, you **must use both**.

Label:

fieldname Name given to the field being defined or modified by the $RSUF macro. It must be 1-8 alphanumeric characters and the first character must be an alphabetic or national character.

Since this name is equated to an assembler symbolic label in several programs (including all RSVP exits), take care to ensure there is no conflict with existing names. All names beginning with $RS are restricted and all names currently used in the $RSCOM common area are restricted. Most other restrictions come from labels in the user exits.

Action:

ADD The field named has not been previously defined and is to be added. An error occurs if the field name given was previously defined.

CHG The field named is to have values changed only for those parameters specified on this $RSUF macro command. An error occurs if the field name was not previously defined.

REP The field named is to be totally replaced by the values given to the parameters on this $RSUF macro command. If a field name was not previously defined, a warning message is issued and this $RSUF macro command is processed as if ADD was specified. (Those parameters not specified default.)

Parameters:

LENGTH

Length of the field. This can be overridden for each of the locations in which a field may appear (SOURCE, INTER, PRINT, TRANS).

TYPE

Defines the type of field (type of logical block in which the field is defined):

CMD	Field entered in the command. (Defined in the PDL - Parameter Descriptor List - generated from $RSPARS. You cannot change the parse list.)
GBL	Field global to all data sets. (Defined in the $RSCOM common area. You cannot add to the common area.)
DS	Data set field placed in the internally formatted data set block.
EXT	Extent field placed in the internally formatted data set block (presence of an extent field causes multiple data set blocks to be built per data set).
VOL	Volume related field. (Defined in the $RSVOL volume block associated with the data set being processed. Users cannot add to the volume block.)
LOCATE	Additional VSAM fields obtained from a catalog LOCATE request.
IPC	IPC related field.
DATCLASS	Data class field.
MGTCLASS	Management class field.
STORCLASS	Storage class field.
STORGROUP	Storage group field.
VD	SMS Volume Definition record.

SOURCE

When processing VTOCs, this indicates the original source field's location, length and format to allow it to be formatted into the proper internal block. The offset is normally required and is a displacement into an FMT1 or FMT4 DSCB, the $RSCOM common area, or the PDL. The length is not needed unless it is different from the value specified in LENGTH. The format normally defaults to CHAR.

PRINT

Indicates where and how the field is to be printed to the PRINT file or displayed on the terminal. This is not normally defined; the offset is calculated dynamically based on the print list being used. The length is normally the value of LENGTH; the format would normally be CHAR.

INTER

Defines the field in its proper internal block. For data set blocks, the offset is calculated dynamically at execution based on the fields referenced in the print and transaction lists and the sort fields. The value given in LENGTH is assumed unless overridden and the format defaults to character.

TRANS

Defines the field as it is on the transaction record. Again the offset is normally calculated dynamically based on the transaction list used, the length defaults to the value given in LENGTH and the format defaults to character.

FLAG

Indicates flags giving additional information about this field:

DATE	It is to be treated as a date for compare. Also, if the TRANS and/or PRINT keywords did not specify length or format, the date is formatted according to the DATES parameter on the $RSUD parameter.
MULTIPLE	The field is multivalued.
$DEFRAG	The field is reported from IPC $DEFRAG records.
INTEGER	It is to be treated as an integer value for comparison.
FLOAT	Arithmetic is performed by floating point instructions.
TOTAL	A total is to be given for this field. (Must be an integer or floating point number.)

Valid formats for INTER, PRINT, SOURCE, and TRANS are:

CHAR	Character
BIN	Binary integer
PD	Packed decimal
HEX	Hexadecimal data
ZD	Zoned decimal
BINDATE	Binary date (3 bytes: yydddd)
PKDATE	Packed decimal date (3 bytes: yyddds)
JDATE	Julian date (5 bytes: yyddd)
MDY	Gregorian date (8 bytes: mm/dd/yy)
DMY	Gregorian date (8 bytes: dd/mm/yy)
YMD	Gregorian date (8 bytes: yy/mm/dd)
CCHH	Cylinder, head
EXIT	To be filled in by an exit
INDEX	To be filled in by an exit
MASK	To be filled in by an exit
COMP	Computed by an exit

EXIT 1- to 8-character name of the module to be called to format this field. For example, exit routine $RSDAYSD in the BASE SRC library.

INDEX Strings to be filled in if source fields are integers value1 through valueN.

MASK Integer string, label, or hex value that can be used in test under mask (TM) construction. If MASK is specified, you must also specify BITSON/BITSOFF.

BITSON Character string. The result of test under mask (TM) determines which string (BITSON or BITSOFF) is substituted.

BITSOFF Character string. The result of test under mask (TM) determines which string (BITSON or BITSOFF) is substituted.

5.4.4.1 Syntax of $RSUF - Defining Literals

You can use a second form of the $RSUF macro to define literals. Since literals can be defined in a list definition by the $RSUL macro, it is not advisable to define literals directly with a $RSUF macro unless their use is quite frequent. When used, only a limited number of parameters are allowed.

LITERAL

A literal string defining the actual literal to be placed in a transaction record (or less likely, a print line).

TRANS

If used, defines an offset (location) into the transaction record. Only the offset operand is allowed on the TRANS parameter and FLAG=FIXDLOC must also be present.

FLAG=FIXDLOC

Indicates that a transaction offset was specified and that the offset is to be forced (so, at execution time, the $RSVP command does not dynamically override the offset).

Examples:

Example 1

The statement `DIVISION $RSUF ADD,LENGTH=5,SOURCE=(62,2,BIN)` defines a new field named DIVISION. The division number is in the Format-1 DSCB at offset 62, is two bytes long, and is binary. It is to be printed as a five-byte number.

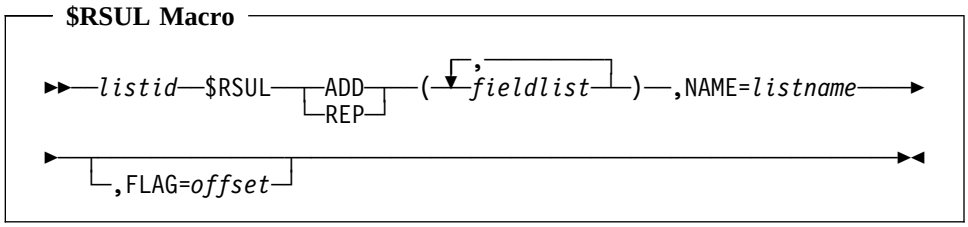
Example 2

The statement `XIN72 $RSUF ADD,LITERAL='X',TRANS=72,FLAG=FIXDLOC` defines a field named XIN72. This field is a literal X and is placed in column 72 of the transaction file for any list definitions that refer to it.

5.4.5 $RSUL Macro

This macro defines lists of fields used for printing, building transaction records, and so on.

Syntax



Syntax Description:

When specifying an action of ADD or REP with the $RSUL macro, you must use NAME=listname. And when you do, you must also use at least one fieldlist.

Label:

listid Assigns an assembler symbolic label to the list. It is only needed if this list is to be referenced by the DEFAULT parameter on a $RSUD macro with FILE=TRANS or FILE=PRINT.

Action

ADD This list has not been previously defined.

REP This list is to replace a previously defined list. If the list indicated by NAME has not been previously defined, a warning is issued and the macro is processed as if ADD were specified.

CHG This is not a valid action for $RSUL macro commands.

Parameter

fieldlist

Indicates a list of fields to be printed, placed on a transaction record (BLIST), or interpreted from an input transaction record (INBLIST). If the list being defined by the $RSUL macro is to be used only for transaction records, multiple field lists can be given to allow multiple transaction records. This could be used to generate multiple commands with a single pass of the VTOC (for instance, to generate IEHPROGM SCRATCH and UNCATLG statements at the same time).

The field list is comprised of any following combination:

fieldname A field name as defined in any previous $RSUF macro command. For multivalued field names, the range of values to be part of the list is expressed as (FIELDNAME, 1:2).

literal A literal string. When a literal is used, it generates an internal call to $RSUF to define it. All previously defined literals are checked first to avoid replication and if the same literal is previously defined, it is not redefined. The literal must be in single quotes.

@nn,'literal' Defines a literal that must be positioned at a specific offset in a transaction record (For example: X in column 72 of any IEHPROGM command that is continued onto a second line). To avoid replication, all previously defined literals are checked for both the same literal and the same offset. If a match is not found, an internal $RSUF macro is generated with TRANS=nn,FLAG=FIXDLOC. (Again, the literal must be in single quotes.)

NAME

The name used in the $RSVP command (LIST, BLIST, INBLIST keywords) to reference this list. It must be one-eight alphanumeric characters.

FLAG=SUPPRESS

Indicates that all fields referenced by the list that are not literals have trailing blanks suppressed from them.

Examples:

Example 1

```
$RSUL ADD,(' SCRATCH VOL=',UNIT,'=',VOLUME,',',@72,'X'),
         (@16,'DSNAME=',DSNAME),NAME=SCRATCH,FLAG=SUPPRESS
```

This statement generates a list with the name SCRATCH to be used in generating IEHPROGM control statements. The literal 'X' is placed in column 72 of the first record and the second record begins in column 16 with 'DSNAME='. ALL nonliteral fields have trailing blanks suppressed.

Example 2

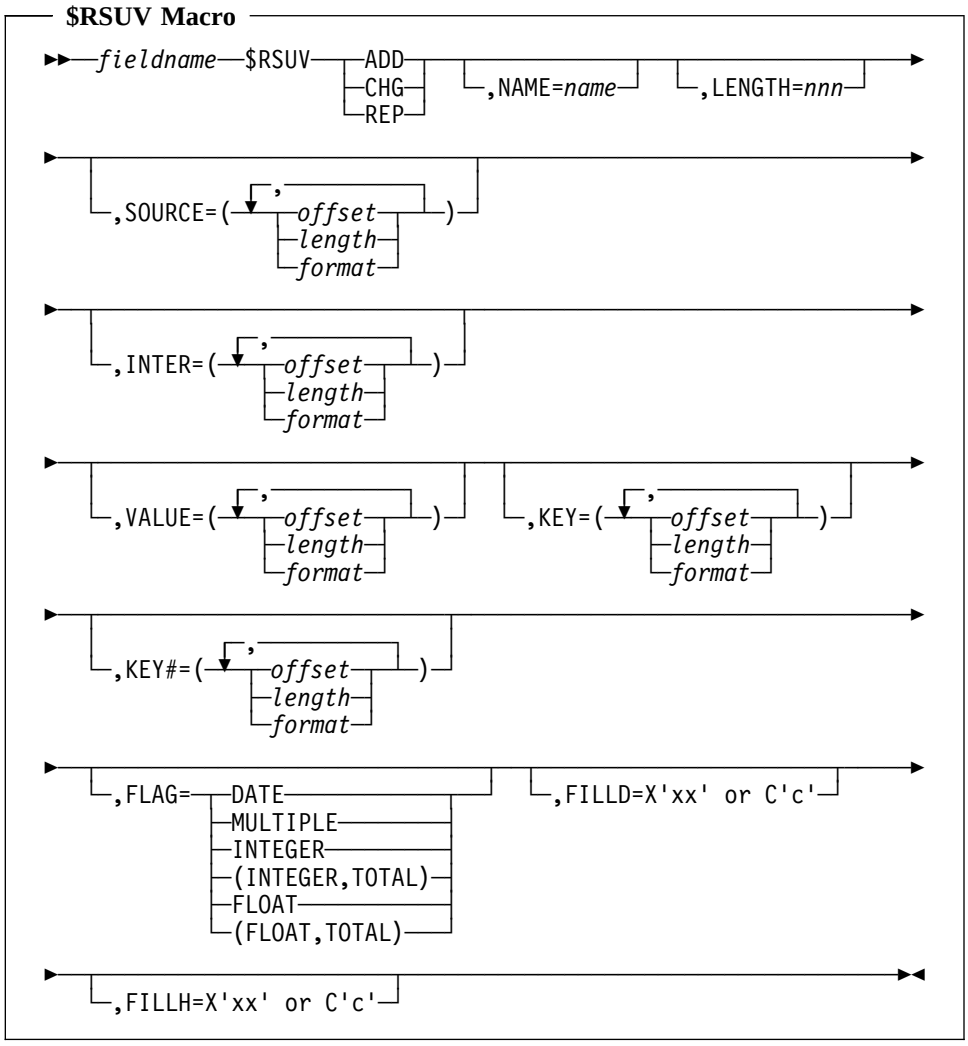
```
$RSUL REP,NAME=VSEXTS                                                  X
         (DSNAME,$EXLOCCH,(EXLOCCH,1:5),(EXHICCH,1:5))
```

This statement replaces list VSEXTS with a new list. Notice how multivalued fields are expressed.

5.4.6 $RSUV Macro

This macro defines how to process the VSAM dictionary entries for a catalog locate request.

Syntax



Syntax Description:

If you specify an action of ADD or REP, you must use the LENGTH=nnn parameter. You can also use any other parameters. Parameters not specified default. Defaults are TYPE=DS, offset=0, length=length specified by LENGTH=nnn above, format=character, and FLAG=no flags.

In all cases, fieldname is required. When specifying FLAG=DATE, you cannot specify other FLAG operands. When specifying the CHG action, you must use at least one parameter.

Label:

fieldname Name given to the field being defined or modified by the $RSUV macro. It must be 1-8 alphanumeric characters and the first character must be an alphabetic or national character.

Since this name is equated to an assembler symbolic label in several programs (including all RSVP exits), take care to ensure there is no conflict with existing names. All names beginning $RS are restricted and all names currently used in the $RSCOM common area are restricted. Most other restrictions come from labels in the user exits.

Action:

ADD The field named has not been previously defined and is to be added. An error occurs if the field name given was previously defined.

CHG The field named is to have values changed only for those parameters specified on this $RSUF macro command. An error occurs if the field name was not previously defined.

REP The field named is to be totally replaced by the values given to the parameters on this $RSUF macro command. If a field name was not previously defined, a warning message is issued and this $RSUF macro command is processed as if ADD was specified. (Those parameters not specified default.)

Parameters:

NAME

Specifies the dictionary field name for a VSAM catalog locate request. Normally, RSVP acquires a number of the entries on its own requiring no additional overhead. If a dictionary field name is specified that is not normally obtained by $VGENDSP, a VSAM locate for that entry is attempted.

LENGTH

Specifies the length of the field. The field length should be the same in the VSAM locate entry and the internal control block (VSL). The length may be specified as

any relocatable expression as a Y type constant. The length specified in the SOURCE or INTER parameter overrides this value.

SOURCE

Specifies the offset into the VSAM locate entry for the data. The source may be specified as any relocatable expression as an A type constant. Optionally, the length and field format may also be specified. The length should be specified if the source and internal lengths of the field are different. The default length is the value of the LENGTH parameter. You only need to specify a format parameter if the source and internal field types are different. The default format is a binary integer.

The $RSUF macro contains a list of valid format types. A format type of MASK is only valid for the SOURCE field. If specified, MASK indicates that this is a 1-byte flag field containing a bit mask. See the VALUE parameter for how the mask is interpreted. Also, for format type of EXIT, see the VALUE parameter for how the exit is invoked. If neither the SOURCE or INTER parameters specify a type, the same type is assumed for each and no conversion is performed.

INTER

Specifies the offset into the internal control block (VSL). If your site wants to modify the RSUSER CSECT by adding additional $RSUV entries, the offset must always be expressed as INTER=VSLUOFF+value. This is because the user portion of the VSL begins at VSLUOFF. You may specify INTER as any relocatable expression expressable as an A type constant.

VALUE

Specifies how the source mask byte should be interpreted or which exit should be invoked. For format type MASK, the mask portion of this parameter specifies any bit mask suitable for a TM instruction. The string1 parameter specifies the character string moved to the internal field if the test set condition code 3 (Branch on Ones). The string2 field parameter specifies the character string moved to the internal field if any other condition code is set. String fields need only be delimited by quote marks if the string contains a blank. For format type EXIT, the string1 parameter field is the name of a module that should be loaded and executed in order to process this field. The string2 parameter may be used to pass parameters to this routine.

KEY

Specifies the keyword to which the field corresponds. The parameter is required.

KEY#

For multivalued keywords, specifies the keyword of the count or # sign field name.

FLAG

Indicates flags giving additional information about this field:

INTEGER It is to be treated as an integer value for comparison.

FLOAT Arithmetic is performed by floating point instructions.

TOTAL A total is to be given for this field. (Must be an integer or floating point number.)

DATE It is to be treated as a date for compare. Also, if the TRANS and/or PRINT keywords did not specify length or format, the date is formatted according to the DATES parameter on the $RSUD parameter.

MULTIPLE The field is multivalued.

FILLD

Specifies the default fill character if an invalid dictionary field name is specified or no data was returned. The default is hexadecimal zeros (X'00'). The parameter may be specified as either X'xx' or C'c' where xx are any two hexadecimal values and c is any valid character.

FILLH

Specifies the default fill character to be returned if the VSAM locate returns hexadecimal FFs for the entire dictionary entry. The default is hexadecimal zeros (X'00'). The parameter may be specified as either X'xx' or C'c' where xx are any two hexadecimal values and c is any valid character.

Example

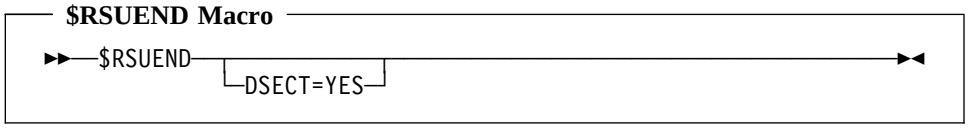
```
VUSER1   $RSUV ADD,CNTREPNO,INTER=(VSLUOFF+4),FLAG=RUVFDATA,        +
               LENGTH=2,KEY=RELREPNO,SOURCE=0
```

This macro causes the RSVP catalog locate process to acquire the CNTREPNO field dictionary entry. The 2-byte field at offset VSLUOFF+4 into the VSL is filled with the first two bytes returned for this dictionary entry. The VSL is an internal control block. Having filled in the VSL field, the user CSECT should then contain an entry indicating how the VSL field is reported. The value of the KEY= parameter must match the name of this entry (supplied through the $RSUF macro). An example of an appropriate $RSUF macro for this field would be:

```
RELREPNO $RSUF ADD,SOURCE=(VSLUOFF+4,2,I),LENGTH=6,TYPE=LOCATE,        +
               FLAG=INTEGER
```

5.4.7 $RSUEND Macro

Syntax



Syntax Description:

DSECT=YES is optional, but when you specify it, you cannot use it in defining the user CSECT. However, you can use it in all other programs to generate a DSECT to reference a user CSECT.

The $RSUEND macro is used to generate the actual assembly of the user CSECT. It uses values previously defined by the $RSUD, $RSUF, $RSUL, and $RSUV macros to generate the user CSECT as you modified it. Once specified, no additional $RSUD, $RSUF, $RSUL or $RSUV macros can be specified.

If an exit you are coding needs to interrogate the user CSECT directly (not advisable since most of the fields of interest are either moved to the $RSCOM common area or located by using the $RSF macro), coding a $RSUEND macro with DSECT=YES generates a DSECT for the user CSECT.

5.4.8 Assembling the User CSECT

When assembling the user CSECT, be sure to provide the following requirements:

1. The assembly step parameter includes RENT.
2. The macro libraries for assembly are: prefix.CAIMAC, SYS1.MACLIB and SYS1.AMODGEN, in this order.
3. The link-edit step parameter includes RENT.
4. The user CSECT is linked as a separate module. It does not need to be linked to the $RSVP command's load module.
5. The CSECT is placed in the load library from which the $RSVP command is executed (in other words, if you do not execute CA-ASM2 directly from CAILIB, the user CSECT is copied from CAILIB to your production library).

The following example uses the standard IBM assembly procedure, ASMHCL:

```
//        JOB
//ASMUSER EXEC ASMHCL,PARM.ASM='OBJ,RENT',
               PARM.LKED='RENT,XREF,LIST,LET,MAP'
//C.SYSLIB DD DSN=prefix.CAIMAC,DISP=SHR
//           DD DSN=prefix.CAISRC,DISP=SHR
//           DD DSN=SYS1.MACLIB,DISP=SHR
//           DD DSN=SYS1.AMODGEN,DISP=SHR
//C.SYSIN  DD  ? <point to your user CSECT>
//L.SYSLMOD DD DSN=prefix.CAILIB,DISP=SHR
//L.SYSIN  DD *
   NAME $RSUSER(R) <or your new user CSECT name>
```

5.5 Performance Notes

Performance is a major consideration in the design and implementation of RSVP. The reentrant coding can reduce paging and permit inclusion of the command in the link pack when heavy usage occurs. The numerous control sections share a single area of working storage to minimize GETMAIN activity. The VTOC reading routine uses full track EXCP programming for speed. You can control blocking of the output files to increase speed. Storage is conserved whenever possible. The SORT function is performed in main storage, using a simple hash technique to improve its speed. External sorting may be done if the main storage requirements are too great.

Catalog search gives poor performance. If the index levels being searched for are known to reside on a limited number of volumes or if the total number online is reasonable, it would be faster to do a volume search and use the BEGIN keyword to control data set selection.

Specifying the field CAT (which indicates the catalog status of a data set) causes poor performance. It means the IBM LOCATE macro is issued on each data set before testing IF, AND, ANDIFx, OR, and ORIFx selection (but after testing for DSNAME selection).

Listing VSAM entities by specifying VSAMCAT(ALL) to allow the Master Catalog to be searched for the proper user catalog gives extremely poor performance and results in response times of minutes rather than seconds, depending on your environment. In a test with one Master Catalog and one user catalog with 40 clusters under MVS 3.8 SE2, response time to list the VSAM cluster entities on all volumes was 1.5 minutes.

Reporting or sorting on a VSAM specific field causes a LOCATE to be performed, thus degrading performance. Reporting or sorting on SMS field names can cause slight performance degradation. RSVP's interface with the SMS subsystem has been optimized to only obtain SMS constructs from the subsystem a minimum number of times, but to also conserve working storage requirements.

Specifying LEVEL(*) for IPC searches causes the entire IPC to be read. Avoid doing this unless it is absolutely necessary or unless a small IPC is maintained on your site. If you know the type of record (Defrag or Unload) or the unload record type (Archive or Backup) you want to query, you can use the IPCTYPE and/or IPCUTYPE keywords in conjunction with LEVEL(*) to limit the number of IPC records read.

For volume searches, BEGIN and LEVEL are the most efficient and provide the best performance. LIKE is the least efficient. Whenever possible, use BEGIN and LEVEL with the other data set selection keywords (CONTAIN, ENDING, and so forth) instead of LIKE. For example, specify: BEGIN(MURTO01) ENDING(DATA) instead of LIKE(MURTO01.-DATA).

In summary, the $RSVP command gains much of its performance by doing only what is required. Command keywords can avoid the processing for formatting the data, printing it, or sorting it. The catalog search and other keyword items are only

formatted and kept in storage if they are requested for printing, sorting, or limits. The catalog search, which uses LOCATE, is very slow. Avoid the catalog search whenever possible. The sorting process takes the majority of the CPU time for large numbers (over 1000) of data sets. If you can use unit, data set name sort, your runtimes decrease significantly. If sorting is not important, turn it off with the NOSORT keyword. By experimenting, you find the combinations of keywords that do what you want most efficiently.

Appendix A. Keywords and Fields

This appendix lists and defines all the keywords and fields you can use when executing RSVP. It includes keywords and corresponding fields from the ICF catalog for VSAM data sets, the IPC, VMF, TMC and the SMS database. The keyword ASM2CAT directs the search to the IPC, the keyword TLMSVMF directs the search to the VMF, and the keyword CA1TMC directs the search to the TMC. For all multivalued fields, RSVP returns a second keyword indicating the total number of values within a field. This field name is the multivalued field name with a # sign prefixing the first character of the name. For field names that are already 8 characters long, RSVP drops the last character. You must request the # sign variable explicitly. The # sign variable is not returned when you specify only the basic variable name.

Observe that for the VSAM and CA-ASM2 catalog fields, blanks are returned if it does not find the cell that contains the information.

Defaults are underlined. The defaults in your system may differ if they were changed in the user CSECT. Please consult your systems programmer for a list of the defaults in your system.

A.1 Keyword List

ACTION('text')
AND1 - AND10(field oper value)
ANDIF1 - ANDIF10(field oper value)
ASM2CAT
AUXFLDS
BEGIN(qualifier)
BLIST(field list)
BOTHEXITS
BREAK(integer)
CATID(catid)
CA1TMC
CHARS(one or two integers)
CLUSTER
COMMENT('text')
CONTAINING(qualifier)
CYLS
DATA
DATASPACE
DATCLASS
DDNAME(ddname)
DSNLEN(integer)
ENDING(qualifier)
EXIT1
EXIT2
EXTSORT
HEADING1('text')
HEADING2('text')
IF(field oper value)
INBILL(ddname)
INBLIST(field list)
INDEXTOTALS(integer)
IPCCOML
KBYTES
LASTLEVEL(index)
LEVEL(index/catindex)
LIKE(dsnmask)
LINES(lines-per-page)
LIST(field-list-name)
MBYTES
MERGE
MGTCLASS
MOUNT(unit)
NEWDATCLASS
NEWMGTCLASS
NEWMGTCLASS
NOEXITS

NOHEADING
NOMERGE
NOSORT
NOTBEGIN(qualifier)
NOTCONTAIN(qualifier)
NOTENDING(qualifier)
NOTLAST(index)
NOTLEVEL(index)
NOTLIKE(dsnmask)
NOTOTALS
NOTVOL(volser)
ORIF1 - ORIF10(field oper value)
OR1 - OR10(field oper value)
OSCATALOG
PERMANENT
PRINT(operand (fields))
QNAME(qname)
QUALIFY(dsnindx)
RETPD(nnnn)
SMSNMLEN
SORT(fields)
STGROUP
STORGACD
STRCLASS
SUBCOMMAND(command (operands))
SUBTOTALS(integer)
TLMSVMF
TOTALS(integer)
TRKS
UGROUP
USERCSECT(module-name)
VOLUME(volser)
VSAMCAT(catname/ddname/ALL)

Although the user CSECT determines the defaults for these keywords, the following defaults were established for the shipped version of RSVP.

BLIST(BILLLIST)
DATASPACE
DDNAME($RSOUT)
KBYTES
LIST(SPACE)
MERGE
PRINT
SORT(DSNAME)
TOTALS
USERCSECT($RSUSER)

The default HEADING1 contains the entered command. Default input is the user's prefix/*logonid*. This is processed by a catalog search.

A.2 Keywords by Category

A.2.1 Search

ASM2CAT	INBLIST	NEWMGTCLASS	STORGACD
CATID	LEVEL	NEWSTRCLASS	STRCLASS
CA1TMC	MGTCLASS	NOTVOL	TLMSVMF
DATCLASS	MOUNT	OSCATALOG	UGROUP
INBILL	NEWDATCLASS	STGROUP	VOLUME
INBLIST			

A.2.2 Selection

By Data Set Name

BEGIN	LASTLEVEL	NOTBEGIN	NOTLAST
CONTAINING	LEVEL	NOTCONTAIN	NOTLEVEL
ENDING	LIKE	NOTENDING	NOTLIKE

Boolean

ANDx	IF	ORx
ANDIFx	ORIFx	

A.2.3 Print

BREAK	HEADING2	LIST	NOTOTALSPRINT
CHARS	INDEXTOTALS	NOHEADING	SUBTOTALS
HEADING1	LINES	NOPRINT	TOTALS

A.2.4 Output

ACTION	DDNAME	NOEXITS	SORT
AUXFLDS	DSNLEN	NOMERGE	SMSNMLEN
BLIST	EXIT1	NOSORT	SUBCOMMAND
BOTHEXITS			
CLUSTER	EXIT2	PERMANENT	TRKS
COMMENT	EXTSORT	QNAME	USERCSECT
CYLS	IPCCOML	QUALIFY	VSAMCAT
	KBYTES		
DATA	MBYTES	RETPD	
DATASPAC	MERGE		

A.3 Keyword Definitions

An alphabetical list of keywords and their definitions follow:

ACTION Defines a character string to be inserted with other fields as part of the records written to the transaction file. The operand must be enclosed in single quotes. For example: ACTION('character string').

ANDx A maximum of ten expression extenders (AND1, AND2, ... AND10) can follow the IF statement. These keywords are tested and allow a data set to be selected only if prior tests were also true. Example: AND1(LSTUS LT 01/01/93). The total number of AND, OR, ANDIF, and ORIF keywords that you can use is ten. At each level (1-10) these keywords are mutually exclusive. This means that if you specify OR1, you cannot specify AND1, ANDIF1, or ORIF1.

ANDIFx A maximum of ten expression extenders (ANDIF1, ANDIF2, ... ANDIF10) can follow the IF statement. ANDIFx is a compound operator used to begin a new logical set. The condition specified on the ANDIFx and any subsequent ANDx or ORx keywords that follow forms the value of this logical set. For a data set to be selected, the value of this logical set and the evaluation of all prior sets must be true.

The total number of AND, OR, ANDIF, and ORIF keywords you can use is ten. At each level (1-10), these keywords are mutually exclusive. This means if you specify OR1, you cannot specify AND1, ANDIF1, or ORIF1.

ASM2CAT Specifies the IPC is to be used as the primary source for input. RSVP searches the IPC and DASD volumes or catalogs. If no catalog identifier (CATID) is specified, the catid $OPT is used. $OPT specifies that fields in the $OPTIONS module be used to define which IPC is to be searched.

$DEFRAG volume records are queried from the IPC if, and only if, a $DEFRAG field is requested. Otherwise, only unload records are returned. Also, when the IPCVTYPE field is specified in a list or on the command line, RSVP returns IPC VSAM path records and IPC unload data set records. For path records, IPCVTYPE contains the value R. The DSNAME field is the name of the path. ASSOCC and ASSOCG are also valid for path records. The value of any other non-IPC type fields are unpredictable for path records. Queries that do not access the IPCVTYPE field can only search IPC unload records.

AUXFLDS Defines a list of auxiliary fields to be processed by the $RSVP command, in addition to all the fields specified or defaulted for LIST (report field list), PRINT (printlist), BLIST (billing field list), and IF (attribute selection).

For space and overhead reasons, RSVP by default only captures and formats those primary fields needed to process the command itself. User exits may need additional fields for other purposes, and those auxiliary fields can be specified by the AUXFLDS keyword.

BEGIN Specifies the character strings with which data set names must begin. To be selected, a data set name must begin with the string specified. Example: BEGIN($CAISYS1.A). Inverse NOTBEGIN.

BLIST Specifies the name of an existing, stored field list used to build output transactions. These field lists were predefined in the user CSECT. The default is BILLLIST. ADAM101 through ADAM610 reports (CA-3 reports) have been built into RSVP to provide even more report options. To use these reports, the appropriate keywords (VOL, IF, LIKE, and so on) must also be specified.

List	Fields
ACTION1	ACTION, DSNAME, COMMENT
ACTION2	ACTION, VOLUME, COMMENT, DSNAME
MIGRATE	DSNAME, VOLUME, NEWSTRCL, NEWMGTCL, NEWDATCL
$AR	'$AR,DSNAME, ('',DSNAME,'')VOL(',VOLUME,RPAREN
$BK	'$BK,DSNAME,('',DSNAME,'') VOL(',VOLUME, RPAREN
$SMRLSE	'$SM DSNAME('',DSNAME, '') VOL(',VOLUME,' RLSE'
$SMCOMP	'$SM DSNAME('',DSNAME,'') VOL(',VOLUME,) COMPRESS RLSE PCTSAVE(10)'
DELETE	' SCRATCH VOL=',UNIT,'=',VOLUME,',',@72,'X'@16, 'DSNAME=',DSNAME'UNCATLG DSNAME=',DSNAME
SCRATCH	' SCRATCH VOL=',UNIT,'=',VOLUME,',',@72,'X'@16, 'DSNAME=',DSNAME
ALLOC	DSCOUNT, ALLOC, USED, UNUSED, PCT, EXT, SECQ, SECT, DSNAME, VOLUME, DSORG, RECFM, BLKSZ, LRECL, CREDT, LSTUS
BILLLIST	DATE, UNIT, VOLUME, DSNAME, ACCOUNT, KBALLOC, KBDAYS, CHCUR

CHARGDET	ACCOUNT, DSCOUNT, KBALLOC, KBDAYS, CHCUR, CHMTD, CHYTD, UNIT, VOLUME, DSNAME
CHARGES	ACCOUNT, DSCOUNT, ALLOC, KBDAYS, CHCUR, CHMTD, CHYTD
COPYCAT	TVOLSER
HISTORY	DATE, CHYTD, CHMTD, CHMONTHS, ACCOUNT
MAPVOL	DSCOUNT, DSNAME, ABSTR, ABLEN, ALLOC, CCHH, EXTSEQ, EXT, VOLUME
SPACE	DSCOUNT, DSORG, LSTUS, ALLOC, USED, UNUSED, EXT, VOLUME, DSNAME

BOTHEXITS Specifies that both the Selection exit ($RSEXT1) and the Post-Sort exit ($RSEXIT2) are to be used. BOTHEXITS is the default.

BREAK Forces page breaks when the primary sort field changes. The format of BREAK is identical to SUBTOTALS. Example: SORT(VOLUME DSNAME) BREAK causes a page break when the volume changes (after the subtotal has been printed if SUBTOTALS was also specified) while SORT(DSNAME) BREAK(4) causes a page break when the first four characters of the DSNAME changes. If INDEXTOTALS is specified, the break is on the high-level index. The $RSOUT DD statement must be present to obtain the break.

CATID Specifies which IPC is to be searched. If no catalog identifier is specified, CA-ASM2 uses the CATID $OPT. $OPT specifies that fields in the $OPTIONS module are used to define which IPC is to be searched. CATID is usually only needed when the IPC is the primary input source. However, CATID may also be required when the field LASTBKP is specified and volumes or system catalogs are the primary input source.

CA1TMC Specifies that the CA-1 Tape Management Catalog (TMC) is to be used as the primary input source. When this keyword is specified, the TMC must be allocated with the DDNAME of TAPEDB prior to the issuing of the command.

Note: CA1TMC and TLMSVMF are mutually exclusive. They cannot be specified on the same $RSVP command.

CHARS Defines the logical record length (number of characters per line) and the block size. The default logical record length is 133 and the default block size is 1330. Both can be changed in the user CSECT. The report output file $RSOUT is record format FBA.

You can specify either one value, or two values separated by blanks, for CHARS. If one value is specified, it is used for both

record length and block size. If two are specified, the first is record length and the second is block size. If you are consistently using a line size other than 133, change the user CSECT to improve blocking. Example: CHARS(121) generates a line size of 120 (1 control character and 120 print positions), and CHARS(121 12100) generates a line size of 120 and a block size of 12100.

CLUSTER Specifies that VSAM elements are to be reported at the cluster level. The keyword VOLUME must be specified with CLUSTER for the USED field on the report to represent the space containing VSAM data. If VOLUME is not included, the USED field represents allocation tracks for VSAM data sets and the UNUSED field is always 0. The field VS contains:

C Cluster entity
G Alternate index
M Master catalog
U User catalog
P Page space
S Swap space

COMMENT Defines a character string that you may include as part of the output records written to the transaction file. The operand must be enclosed in single quotes. For example: COM('character string').

CONTAINING Specifies qualifiers that can be located anywhere within the data set name and that can cause the data set to be selected. For example: CONTAINING(MAC ARCH.$). Inverse NOTCONTAIN.

CYLS Allocation is in cylinders. Because of the differences in track sizes on different devices, CYLS is not meaningful on mixed devices, and rounding limits its usefulness for small data sets.

DATA Specifies VSAM elements are to be reported at the data or index level. When DATA is specified, the field VS contains either D for data elements or an I for index elements. The VOLUME keyword must be specified with DATA for data/index level reporting.

DATASPACE Specifies that VSAM data sets are to be reported only by the information in the Format-1 DSCB in the VTOC. DATASPACE is useful for identifying VSAM components quickly. DATASPACE is the default. The field VS is blank.

DATCLASS Indicates that SMS Data Classes are to be searched.

DDNAME Specifies the report DDNAME. The default is $RSOUT. The default can be changed in the user CSECT. All references in this text assume the default of $RSOUT.

DSNLEN Specifies the size of the DSNAME field on the report and the output transaction file ($RSTRANS). The default is 44. Another default can be established in the user CSECT. Example: DSNLEN(20) truncates the data set name to 20 positions.

ENDING Specifies the character strings with which data set names must end. Example: `ENDING(LIST A.DATA)` Inverse NOTENDING.

EXIT1 Causes only the first user exit (Selection exit $RSEXT1) to execute.

EXIT2 Causes only the second user exit (Post-Sort exit $RSEXIT2) to execute.

EXTSORT Valid only if INBILL is specified. This implies the data was presorted in the order specified in the SORT keyword by a previous step. This avoids an in-core sort, but allows other keywords such as BREAK, SUBTOTALS, or INDEXTOTALS to function. This is useful and sometimes necessary when large amounts of data are involved, such as a billing history accumulation run.

HEADING1 Replaces the normal heading on a report. The heading must be in single quotes and the first position contain an ASA print control character. A 1 in the first position indicates the HEADING1 line is to be on the top of a new page. Spacing is exactly as entered. For example: HEADING1('1 MY REPORT') causes a page break at the heading, and the report is titled MY REPORT. You must include the $RSOUT DD statement to obtain this heading.

HEADING2 Defines a second title line. The format is identical to that of HEADING1. For HEADING2, the current date is printed on the second title line below the page number. Example: HEADING2('- BY JOE BLOE') causes the second line to be spaced down 3 lines from the first. It prints BY JOE BLOE followed by the current date. You must include the $RSOUT DD statement to obtain this heading.

IF Specifies the first test for data set selection. It must be present for any ANDx, ORx, ANDIFx, or ORIFx to be interpreted. Example: IF(DSORG EQ PO)

INBILL Defines the DDNAME of an existing transaction file from a previous $RSVP command. (The previous command allocated this data set with the DDNAME $RSTRANS.) This DDNAME must not be $RSTRANS for input since RSVP would open the $RSTRANS file for output while trying to read it for input. The same data set cannot be allocated to both DDNAMEs.

INBLIST Defines the field list used to create the existing transaction file. If you do not specify an operand, the default is the list specified in BLIST. If BLIST is not specified, the default is BILLLIST. The default can be changed in the user CSECT. See BLIST and LIST for a complete set of field lists.

INDEXTOTALS Provides subtotals on the high-level qualifiers of the DSNAME. INDEXTOTALS is only valid if DSNAME is the first sort field. If not, SUBTOTALS is assumed instead of INDEXTOTALS. The default is 1, indicating a subtotal on the first index level. You can enter a value of 1, 2, 3, or 4 to indicate the index level on which subtotaling is to be done. Example: SORT(DSNAME) INDEXTOTALS gives subtotals each time the high-level qualifier changes while SORT(DSNAME) INDEXTOTALS(2) subtotals whenever the second-level qualifier changes and when the high-level qualifier changes.

IPCCOML Indicates the length of the comment field that is reported if the IPCCOM field is part of a print or billing list. The default is 30 characters.

KBYTES Allocation is in kilobytes (thousands of bytes). KBYTES shows reasonable accuracy; it allows data sets on different device types to be represented in the same terms. KBYTES is the default. The default can be changed in the user CSECT.

LASTLEVEL Selects data sets with the specified low-level indexes. Example: LASTLEVEL(ASM CNTL $RS.ASM). Inverse NOTLAST.

LEVEL Selects data sets with specified high-level indexes, for example: LEVEL($CAI03 $CAI01.$RS). LEVEL can specify specific high-level indexes to be used in the catalog search such as LEVEL($CAI01 $CAI03). Inverse NOTLEVEL.

LIKE Specifies the data set name pattern masks on which data sets are selected. Example: LIKE($CAI**.-.ASM $CAI**.-.$RS-.-). See Pattern Masking on page 1-6 for more information and examples. The inverse of NOTLIKE.

LINES Defines the number of lines per page. The default is 60 but you can change it in the user CSECT. Example: LINES(80) sets the number of lines per page to 80.

LIST Specifies the name of a stored report field list. These field lists were predefined in the user CSECT. Example: LIST(USAGE). The default is SPACE. ADAM101 through ADAM610 reports (CA-3 reports) have been built into RSVP to provide even more report options. To use these reports, the appropriate keywords (VOL, IF, LIKE, and so on) must also be specified. Valid lists of the fields to be printed are:

List	Fields to Print
ADAM101	VOLUME, DSNAME, DSORG, EXT, CCHH, SECT, SECQ, ALLOC, USED, PCT, DSCOUNT
ADAM102	VOLUME, DSNAME, STRACK, ETRACK, TRKALLOC, CCHH, FCYLS, EXTSEQ, EXT, DSCOUNT

ADAM102V	VOLUME, VOLSPC, VOLALLOC, VOLIXSP, VOLEXT
ADAM103	VOLUME, DSNAME, EXT, DSORG, RECFM, BLKS2, LRECL, CREDT, EXPDT, LSTUS, PW, CAT, DSCOUNT
ADAM104	VOLUME, DSNAME, EXT, DSORG, RECFM, BLKS2, LRECT, CREDT, EXPDT, LSTUS, DSCOUNT
ADAM105	VOLUME, DSNAME, CREDT, EXPDT, LSTUS, LMDATE LMJOB, TRKALLOC, DSCOUNT
ADAM 201	VOLUME, FRCYLS, FRTRKS, LFCYLS, LFTRKS, VOLEXT
ADAM301	VOLUME, DSNAME, CREDT, EXPDT, LSTUS, LMDATE, ALLOC, DSCOUNT, CB
ADAM610	DSNAME, VOLUME, UNIT, UNLVOL, UNLFSEQ, CDATE, EDATE, IPCDATE, UNLRETPD, C3DSORG, BLKSIZE, ALLOC
	DSCOUNT, ALLOC, USED, UNUSED, PCT, EXT, SECQ, SECT, DSNAME, VOLUME, DSORG, RECFM, BLKSZ, LRECL, CREDT, LSTUS
BILLLIST	DATE, UNIT, VOLUME, DSNAME, ACCOUNT, KBALLOC, KBDAYS, CHCUR
CHARGDET	ACCOUNT, DSCOUNT, KBALLOC, KBDAYS, CHCUR, CHMTD, CHYTD, UNIT, VOLUME, DSNAME
CHARGES	ACCOUNT, DSCOUNT, ALLOC, KBDAYS, CHCUR, CHMTD, CHYTD
HISTORY	DATE, CHYTD, CHMTD, CHMONTHS, ACCOUNT
MAPVOL	DSCOUNT, DSNAME, ABSTR, ABLEN, ALLOC, CCHH, EXTSEQ, EXT, VOLUME
SPACE	DSCOUNT, DSORG, LSTUS, ALLOC, USED, UNUSED, EXT, VOLUME, DSNAME
UNLOAD	IPCDATE, IPCUTYPE, IPCHVOL, IXRCAND, UNLVOL, UNLFSEQ, DSNAME
USAGE	DSCOUNT, USECNT, LSTUS, LMDATE, LMTIM, LMJOB, VOLUME, ALLOC, DSNAME

VOLUME	VOLUME, CUU, UNIT, DEVTYPE, VOLATTR, VOLSPC, VOLEXT, VOL1XSP, DSREC, DSTOT, VTOCI, CYLPVOL, TRKPCYL, DEVTK, DEVDT, DEVDB, VTOCE, VTOCL, VOLLSPC
VSEXTS	DSNAME, HIALLRBA, HIUSERBA, #EXLOCEH, EXLOCCH(1:2), EXHICCH(1:2)
VSSTATS	DSNAME, CIPCA, PCA, PCI, NUMRECS, RETRECS
VSSTATSX	DSNAME, CIPCA, PCA, PCI, NUMRECS, DELRECS, INSRECS, UPDRECS, RETRECS, CISPLITS, CASPLITS, NUMEXCPS

If the print line cannot hold all the fields defined in a list (if the print-line length exceeds the CHARS value), ending fields with lengths exceeding 18 (such as DSNAME) are truncated and shorter fields are ignored. Other print lists can be defined by changes to the user CSECT (see 5-9 in the "Tailoring" chapter).

MBYTES Allocation is in megabytes (millions of bytes).

MERGE Causes the output from all primary input sources to be combined into one report. NOMERGE divides the output into sections based on the primary sources of input. A sort is done separately for input source. MERGE may require a large amount of virtual storage to be acquired. MERGE is the default.

MGTCLASS Indicates that SMS Management Classes are to be searched.

MOUNT Specifies a unit name or device type on which to mount only specific volumes. For example: `MOUNT(3330-1), MOUNT(26C), or MOUNT(SYSMNT)` (This works online only if the MOUNT attribute is specified in SYS1.UADS.)

NEWDATCLASS Specifies a 1- to 8-character SMS Data Class name.

NEWMGTCLASS Specifies a 1- to 8-character SMS Management Class name.

NEWSTRCLASS Specifies a 1- to 8-character SMS Storage Class name.

NOEXITS Eliminates invocation of all exits.

NOHEADING Eliminates all page breaks and heading lines, including the default.

NOMERGE Causes the output to be divided into sections based on the primary input sources. Each input source is sorted separately. Inverse of MERGE.

NOPRINT Eliminates the printing of detail lines. However, the fields valid in the PRINT keyword (ADD, REP, DEL, and NEW) are valid in NOPRINT to allow control of the subtotal and total print lines. NOPRINT causes an additional default of NOSORT.

NOSORT Bypasses the sort phase and allows printing to take place as a data set is selected. This may speed processing. If you specify NOSORT, the report order is unit address and volume. This also nullifies the use of BREAK, SUBTOTALS, and INDEXTOTALS. For a catalog type search, you can specify NOSORT if the level parameters are specified in ascending order.

NOTBEGIN Excluded data sets with the specified high-level qualifier(s). Example: `NOTBEGIN($CAI SYS1.A)`

NOTCONTAIN Excludes data set name containing the specified qualifiers. Example: `NOTCONTAIN(MAC ARCH.$)`

NOTENDING Excludes data sets ending with the specified low-level qualifiers. Example: `NOTENDING(LIST A.DATA)`

NOTLAST Excludes data sets ending with the specified low-level indexes. Example: `NOTLAST(ASM CNTL $RS.ASM)`

NOTLEVEL Excludes data sets beginning with the specified high-level indexes. Example: `NOTLEVEL($CAI03 $CAI01.$RS)`

NOTLIKE Identifies data set name pattern masks that can exclude any data sets matching the patterns. Example: `NOTLIKE($CAI**.-.ASM $CAI**.-.$RS-.-)` See Pattern Masking on page 1-6 in the "Introduction" chapter for more information.

NOTOTALS Eliminates all subtotals and final totals.

NOTVOL Identifies the generic or specific volume serial numbers to be excluded from selection. NOTVOL overrides VOLUME.

ORIFx A maximum of ten expression extenders (ORIF1, ORIF2, ... ORIF10) can follow the IF statement. ORIFx is a compound operator used to begin a new logical set. The condition specified on the ORIFx and any subsequent ANDx or ORx keywords that follow can forms the value of this logical set. A data set is selected if the value of this logical set is true regardless of the value of any previous logical sets.

The total number of AND, OR, ANDIF, and ORIF keywords you can use is ten. At each level (1-10), these keywords are mutually exclusive. This means if you specify OR1, you cannot specify AND1, ANDIF1, or ORIF1.

ORx A maximum of ten expression extenders (OR1, OR2, ... OR10) can follow the IF statement. The results of the expression extenders allow a data set to be selected regardless of all prior tests within the current logical set. The total number of AND, OR, ANDIF, ORIF keywords that you can use is ten. At each level (1-10) these keywords are mutually exclusive. This means that if you specify OR1, you cannot specify AND1, ANDIF1, or ORIF1.

OSCATALOG Specifies to perform catalog searches on the high-level indexes specified by the LEVEL keyword. This option lets you search catalogs and other primary sources with the same invocation.

PERMANENT Places the data set on a permanent archive tape.

PRINT Modifies or replaces the print list. For multivalued field names, you can express a range of values to be part of the list by using a colon to separate the field values. For example, `HIALLRBA(1:5)` prints the first through fifth values of HIALLRBA.

There are four options for using the PRINT keyword:

ADD Add fields to the print list. Following ADD, enter a list of fields. The first field entered must be a field that is in the active print list. The additional fields listed are inserted after this field. Example: `PRINT(ADD (LSTUS LMDATE LMTIM LMJOB))` inserts last modified date (LMDATE), time (LMTIM), and job (LMJOB) into the print list following last use date (LSTUS).

REP Replace a field in the print list with other field(s). The format is the same as ADD, but the first field listed is deleted from the list. Example: `PRINT(REP (LSTUS LMDATE LMTIM LMJOB))` replaces last use date (LSTUS) with last modified date(LMDATE), time (LMTIM), and job (LMJOB).

DEL Deletes fields in the current print list. If only one field is being deleted, it is not necessary to put it into parentheses, but multiple fields must be in parentheses. Example: `PRINT(DEL DSCOUNT)` deletes the data set count while PRINT(DEL (DSCOUNT UNUSED LSTUS)) deletes the data set count (DSCOUNT), the unused space (UNUSED), the last used date (LSTUS).

NEW Ignores the print list and creates a new print list using the fields in parentheses. If there is only one field, it is not necessary to put it into parentheses, but multiple fields must be in parentheses. Example: `PRINT(NEW (ALLOC PCT DSORG DSNAME))` generates a report with the fields: ALLOC (allocation), PCT (percent used), DSORG (data set organization), and DSNAME (data set name).

Note: A blank must follow the words ADD, REP, DEL, NEW.

QNAME Places the requests in a logical queue of requests with the qname specified. Data sets archived (or backed up) with the same qname are unloaded onto the same tape.

QUALIFY Renames the specified data sets by adding the dsnindx to the existing name when archived (or backed up). Restoration must be under the new name. The dsnindx is a 1- to 8-character string.

RETPD Specifies the number of days (up to your installation-set value) to retain the data on tape. A value of 9999 sets RETPD to your installation-defined value.

SMSNMLEN Specifies the length of the SMS construct name:

DATCLASS
MGTCLASS
STGROUP
STRCLASS

The default is eight characters. The maximum length is 30 characters.

SORT Determines the sort order. By default, sort is in ascending order. For example, SORT(VOLUME DSNAME) sorts first on VOLUME then on DSNAME. To sort in descending order, place a minus sign in parentheses after the selected field. For example, `SORT(DSNAME DSORG(-))`.

To sort on multivalued fields, place the number of the value you want to sort on in parentheses after the field. For example, HIALLRBA(4) sorts (in ascending order) on the fourth value for the HIALLRBA field. To indicate a descending sort for a multivalued field, put a minus before the indicated value. For example, `HIALLRBA(-4)` sorts on the fourth value in descending order.

STGROUP Identifies the SMS Storage Group(s) that are searched. Storage Group names can be specified with up to 30 characters. Pattern masking is supported.

STORGACD Indicates that the SMS Storage Groups are to be searched. STRGROUP is supported as an ALIAS of STORGACD.

STRCLASS Indicates that the SMS Storage Classes are to be searched.

SUBCOMMAND Specifies whether $AR or $BK commands are being created. These commands place the identified data sets in the archive or backup queue, as appropriate. When the queues are processed, the data sets are actually archived or backed up. Example: `SUBCOMMAND($AR)` or `SUBCOMMAND($BK)`

SUBTOTALS Causes subtotaling to take place whenever the primary sort field changes. An integer in parentheses following SUBTOTALS indicates the number of meaningful characters in the primary sort field. Whenever these characters change, subtotaling is done. Example: `SORT(VOLUME DSNAME) SUBTOTALS` gives a subtotal each time the volume changes while SORT(DSNAME) SUBTOTALS(4) gives a subtotal every time the first four characters of DSNAME change.

TLMSVMF Specifies that the TLMS Volume Master File (VMF) is to be used as the primary input source. When this keyword is specified, the VMF must be allocated with the DDNAME of CAIVMF prior to issuing of the command.

Note: CA1TMC and TLMSVMF are mutually exclusive. They cannot be specified on the same $RSVP command.

TOTALS Similar to SUBTOTALS, it causes totaling to take place whenever the primary sort field changes. An integer in parentheses following TOTALS indicates the number of meaningful characters in the primary sort field. Whenever these characters change, totaling is done. Example: SORT(VOLUME DSNAME) TOTALS gives a total each time the volume changes while SORT(DSNAME) TOTALS(4) gives a total every time the first four characters of DSNAME change. TOTALS is the default.

TRKS Allocation in tracks. Because of differences in track sizes on different devices, TRKS is not an accurate measurement when combining mixed device types.

UGROUP Identifies the non-SMS esoteric unit(s) that are searched.

USERCSECT Specifies the load module name of the user CSECT for this execution of the command. The default is $RSUSER. Example: `USERCSECT(MYCSECT)`

VOLUME Identifies the following volumes to search:

1. Specific volumes by the full 6-character volume serial number, such as VOLUME(SYS001 SYS005 WORK01 WORK02).
2. Generic volumes beginning with a 1- to 6-character string, such as VOLUME(SYS WORK).
3. All volumes, as referenced by an asterisk (*), such as VOLUME(*).

VSAMCAT Specifies which catalogs are to be searched for VSAM statistics for the VSAM components found on the volumes being searched. If VSAMCAT is not specified, RSVP searches the default catalog associated with the JOB step. For VSAMCAT, the values are:

1. A VSAM catalog name: VSAMCAT(USER.CAT.NAME).
2. The name of a DD that defines the VSAM catalog to be searched: VSAMCAT(CATDD).
3. The keyword ALL which means to search all VSAM catalogs that can be found in the system where the CA-ASM2 command processor is running: VSAMCAT(ALL).

However, avoid using VSAMCAT(ALL) if possible. It creates extremely poor performance and may result in response times of minutes rather than seconds, depending on your environment. See Performance Notes on page 5-33 in the "Tailoring" chapter for more information.

For example, VSAMCAT(CATALOG.USER1) CLUSTER reports on clusters on the volumes specified as long as they are in CATALOG.USER1. The statement

```
//CATDD DD DSN=CATALOG.USER1,DISP=SHR
```

with VSAMCAT(CATDD) DATA lists the data and index elements of VSAM data sets on the specified volumes as long as they are in CATALOG.USER1.

When listing VSAM, the field BLKSZ can have CISIZE and the field LRECL can have maximum record length.

A.4 Fields by Category

The following fields that you can use in $RSVP commands are grouped by function. The default output field length is in parentheses following the field name.

A.4.1 Data Set Fields

ADDRESS (10)
ALLOC (8)
BLKSZ (5)
C3DSORG (3)
CAT (3)
CB (2)
CDATE (5)
CREDT (8)
DSIND (5)
DSNAME (44)
DSORG (5)
EDATE (5)
ETRACK (8)
EXPDT (8)
FCYLS (6)
INUSE (3)
KBALLOC (8)
KEYL (4)
LDATE (5)
LMDATE (8)
LMJOB (8)
LMTIM (5)
LRECL (5)
LSTUS (8)
LV (2)
MDATE (5)
MODID (8)
OPTCD (5)
PCT (3)
PW (2)
RACF (4)
RD (2)
RECFM (5)
RETPD (6)
RKP(4)
SECQ (5)
SECT (4)
STRACK (8)
SUL (3)
TRKALLOC (8)
UNUSED (8)
USECNT (8)
USED (8)
VOLSQ (5)
VS (2)

A.4.2 Data Set Extent Fields

ABLEN (5)
ABSTR (5)
CCHH (17)
EXT (5)
EXTSEQ (6)

A.4.3 Volume Fields

CYLPVOL (7)
CUU (4)
DEVDB (5)
DEVDT (5)
DEVTK (5)
DEVTYPE (8)
DSREC (5)
DSTOT (6)
FRCYLS (6)
FRTRKS (6)
LFCYLS (6)
LFTRKS (6)
TRKPCYL (7)
TRKPVOL (8)
UNIT (8)
VOLALLOC (8)
VOLATTR (8)
VOLCAP (8)
VOLEXT (6)
VOLLSPC (30)
VOLNDX (6)
VOLPOOL (8)
VOLSPC (8)
VOLUME (6)
VOL1XSP (8)
VPCT (4)
VTOCE (17)
VTOCI (5)
VTOCL (6)

A.4.4 Billing Fields

ACCOUNT (10)
BILLDAYS (6)
BILLHRS (6)
CHCUR (8)
CHMONTHS (96)
CHMTD (8)
CHYTD (8)
KBDAYS (9)
KBHRS (11)
TRKDAYS (8)
TRKHRS (8)

A.4.5 Miscellaneous Fields

ACTION (79)
COMMENT (40)
DATE (8)
DSCOUNT (7)
ERROR (16)
EXITFLD (8)
SC (2)
SOURCE (8)
TIME (8)
TYPE (4)
WEEKDAY (9)
YDD (5)

A.4.6 Predefined Literal Fields

LPAREN (1)
L$ARDSN (1)
L$BKDSN (1)
L$RLSE (1)
L$RPVOL (1)
QUOTE (1)
RPAREN (1)
SPACE (1)

A.4.7 VSAM Fields

ASSOCC (44)
ASSOCD (44)
ASSOCG (44)
#ASSOCG (6)
ASSOCI (44)
ASSOCR (44)
#ASSOCR (6)
AVGLRECL (10)
#ASSOCR (6)
BUFSIZE (10)
CANDVOL (7)
#CANDVOL (6)
CASPLITS (10)
CIPCA (6)
CISPLITS (10)
DELRECS (10)
ERASE (5)
EXCPEXIT (8)
EXHICCH (8)
#EXHICCH (6)
EXHIRBA (8)
#EXHIRBA (4)
EXLOCCH (8)
#EXLOCCH (6)
EXLORBA (8)
#EXLORBA (6)
EXTRKS (6)
#EXTRKS (6)
HIALLRBA (10)
#HIALLRB (6)
HIALLRDS (8)
HIUSERBA (10)
#HIUSERB (6)
HIUSERDS (8)
IMBED (5)
INDEXLEV (6)
INSRECS (10)
NUMEXCPS (10)
NUMRECS (10)
ORDERED (7)
OVERFVOL (8)
#OVERFVO (6)
OWNERID (16)
PCA (3)
PCI (3)
PHYRSIZE (10)
#PHYRSIZ (6)
PHYRTRK (6)
#PHYRTRK (6)
PHYRTRK (6)
#PRIMEVOL (6)
PRIMEVO (67)
PROTECT (7)
RECOVERY (8)
RELIND (1)
REPLCATE (8)
RETRECS (10)
SHROPTNS (5)
TRKSCA (6)
#TRKSCA (6)
TV (1)
UNIKEYS (7)
UNIQUE (6)
UPDRECS (10)
VEXTS (4)
#VEXTS (6)
VXTYPE (2)
#VXTYPE (6)
WRTCHK (6)

A.4.8 IPC Fields (CA-ASM2)

ASSOCC (44)
ASSOCD (44)
ASSOCG (44)
#ASSOCG (6)
ASSOCI (44)
ASSOCR (44)
#ASSOCR (6)
DFGFILE1 (44)
DFGOMED (4)
DFGOVOL (6)
#DFGOVOL (4)
DIRBLKS (8)
DYNARSN (9)
DYNBRSN (9)
DYNDTYPE (9)
DYNENQ (7)
DYNRC (6)
DYNSYSC (7)
IPCCOM (256)
IPCCSTAT (7)
IPCDATE (5)
IPCDELDT (5)
IPCDELID (8)
IPCHVOL (6)
IPCLDEL (7)
IPCSYSID (4)
IPCTIME (8)
IPCTYPE (1)
IPCUTYPE (7)
IPCVTYPE (7)
IXRCAND (1)
IXRCDVOL (6)
#IXRCDVO (6)
IXRDEVT (8)
IXRFDBK (8)
IXRLDATE (5)
IXRLTIME (8)
IXRRELDA (8)
IXRRELDC (8)
IXXRRLVOL (6)
IXRSYSID (4)
IXRTDSN (44)
RACGDG (6)
RACGPROF (7)
RACIND (6)
RACPNAME (44)
RACPVOL (6)
RLDALRC (6)
RLDCATRC (6)
RLDCOPY (6)
RLDFORCE (5)
RLDIXR (6)
RLDLCRC (6)
RLDRC (6)
RLDRDATE (5)
RLDRENM (7)
RLDRTIME (8)
RLDRVOL (6)
#RLDRVOL (6)
RLDUCC3 (6)
RLDUSER (8)
RLDVDEVT (8)
#RLDVDEV (6)
RLDVIRT (7)
UC3DEN (7)
UC3DESQ (8)
UC3FEET (8)
UC3FSEQ (8)
UC3RUN (7)
UC3VSAM (8)
UC6PSWD (8)
UC6ST (5)
UNLAERR (5)
UNLARCAT (6)
UNLASCRT (6)
UNLAUCAT (6)
UNLBKBV (8)
UNLBLK (8)
UNLBLKSZ (8)
UNLBYTES (12)
UNLBYTKB (8)
UNLCBYTE (12)
UNLCKBC (8)
UNLCOMP (7)
UNLCOPY (4)
UNLCTECH (8)
UNLEMPTY (5)
UNLFORM (8)
UNLFSEQ (6)
UNLFVOL (7)
UNLIBKP (5)
UNLMAMT (12)
UNLMDTYP (6)
UNLMGR (6)
UNLNRBLK (7)
UNLODSN (44)
UNLPCYCL (5)
UNLPERM (7)
UNLRCAT (6)
UNLRCATP (8)
UNLRETPD (8)
UNLSCRT (6)
UNLSYSID (8)
UNLTRKS (8)
UNLUARC (5)
UNLUBKP (5)
UNLUCAT (6)
UNLVERS (4)
UNLVIRT (7)
UNLVOL (6)
UNLV# (6)

A.4.9 SMS Fields

DATALLOC (8)
DATAUNIT (6)
DATCISZ (8)
DATCLASS (30)
DATDATE (10)
DATDAYSD (10)
DATDESC (120)
DATDIBLK (6)
DATEXDAT (10)
DATIMBED (5)
DATKLEN (4)
DATKOLL (6)
DATLRECL (6)
DATPCA (3)
DATPCI (3)
DATRCORG (4)
DATRECFM (3)
DATREPLC (8)
DATRETPD (8)
DATSHROP (5)
DATSPPRI (10)
DATSPSEC (10)
DATTIME (8)
DATUSER (8)
DATVOLCT (4)
DATXREG (2)
DATXSYS (2)
DAYSDEL (10)
DAYSLFTA (10)
DAYSLFTB (10)
DAYSLFTD (10)
DAYSLFTS (10)
LASTBKP (10)
MAXDAYSD (10)
MGTAUTBK (7)
MGTBADU (4)
MGTBKDY (6)
MGTBKFQ (6)
MGTBKNP (6)
MGTBKVS (6)
MGTBVRD (6)
MGTCLASS (30)
MGTCMAU (4)
MGTDATE (10)
MGTDESC (120)
MGTEXDAT (10)
MGTEXDAY (10)
MGTEXPDY (10)
MGTL1DY (6)
MGTMRTDY (6)
MGTPELEM (6)
MGTPRDY (6)
MGTPREL (7)
MGTRLOMG (7)
MGTTIME (8)
MGTUSER (8)
NEWDATCL (8)
NEWMGTCL (8)
NEWSTRCL (8)
SGVCONV (7)
SGVDATE (10)
SGVMVSS1 (8)
SGVMVSS2 (8)
SGVMVSS3 (8)
SGVMVSS4 (8)
SGVMVSS5 (8)
SGVMVSS6
SGVMVSS7 (8)
SGVMVSS8 (8)
SGVSGRP (30)
SGVSMSS1 (7)
SGVSMSS2 (7)
SGVSMSS3 (7)
SGVSMSS4 (7)
SGVSMSS5 (7)
SGVSMSS6 (7)
SGVSMSS7 (7)
SGVSMSS8 (7)
SGVSTAT (32)
SGVSTGST (7)
SGVTIME (8)
SGVUSER (8)
STGABKP (7)
STGABSYS (8)
STGADMP (7)
STGADYSYS (8)
STGAMIG (7)
STGAMSYS (8)
STGDATE (10)
STGDESC (120)
STGDMPCL (40)
STGDMPC1 (8)
STGDMPC2 (8)
STGDMPC3 (8)
STGDMPC4 (8)
STGDMPC5 (8)
STGHTHR (3)
STGLTHR (3)
STGPRST (16)
STGPRST1 (2)
STGPRST2 (2)
STGPRST3 (2)
STGPRST4 (2)
STGPRST5 (2)
STGPRST6 (2)
STGPRST7 (2)
STGPRST8 (2)
STGROUP (30)
STGSTAT (7)
STGTHRSH (9)
STGTIME (8)
STGTYPE (5)
STGUSER (8)
STGVMAX (10)
STGVUNIT (4)
STRAVAIL (4)
STRCLASS (30)
STRDATE (10)
STRDBIAS (5)
STRDESC (120)
STRDRESP (8)
STRGSP (9)
STRSBIAS (5)
STRSRESP (8)
STRTIME (8)
STRUSER (8)

A.4.10 TMC Fields (CA-1)

TAUDDTE (10)
TAUDTIM (8)
TBADIND (1)
TBASVOL (6)
TBLKCNT (8)
TBLKSIZ (5)
TCDATE (10)
TCDDNME (8)
TCJOB (8)
TCLNIND (1)
TCLSIND (1)
TCPUID (4)
TCSTEP (8)
TCTIME (8)
TCUNIT (3)
TDELIND (1)
TDEN (4)
TDSNACT (1)
TDTECLN (10)
TDTEMOV (10)
TDYNIND (1)
TEDMIND (8)
TEXPDTP (5)
TFILCNT (5)
TFILSEQ (5)
TFSTVOL (6)
THD1DSN (17)
TINPUT (1)
TINTCIN (1)
TINTDTE (10)
TLABTYP (3)
TLDATE (10)
TLJOB (8)
TLRECL (5)
TLTIME (8)
TLUNIT (3)
TLUPDAT (8)
TMANDUPD (1)
TNXTVOL (6)
TOOUTIN (1)
TOUTLOC (4)
TOUTPUT (1)
TOUTSLT (6)
TOUTSRV (4)
TPRDCLN (6)
TPRDINT (6)
TPREVOL (6)
TPWTCLN (6)
TPWTINT (6)
TRDCAT (1)
TRDSOVR (1)
TRECFM (4)
TRECIND (1)
TSCRFLG (1)
TSCRING (1)
TTMSCLN (6)
TTPMIND (1)
TTRDCLN (6)
TTRDINT (6)
TTRTCH (4)
TTWTCLN (6)
TTWTINT (6)
TUSECLN (6)
TUSERID (8)
TUSETOT (6)
TUSRDTA (60)
TUSRIND (1)
TVENDOR (8)
TVOLOWN (8)
TVOLSEQ (5)
TVOLSER (6)
TXBYCAT (1)
TXBYCYC (1)
TXBYJUL (1)
TXBYLDT (1)
TXPROC (1)

A.4.11 VMF Fields (CA-Dynam/TLMS)

TACTFIL (4)
TBASVOL (6)
TBLKCNT (8)
TBLKSIZ (5)
TCDATE (10)
TCDDNME (8)
TCDSSEQ (4)
TCJOB (8)
TCLSIND (1)
TCPGM (8)
TCPUID (4)
TCSTEP (8)
TCTIME (8)
TCUNIT (3)
TDATCRT (10)
TDATSCR (10)
TDEN (4)
TDKPDTE (10)
TDKPTYP (2)
TDTECLN (10)
TDTEMOV (10)
TEXPDT (10)
TEXPDTP (5)
TFILCNT (5)
TFILSEQ (5)
TINTDTE (10)
TLABTYP (3)
TLDATE (10)
TLJOB (8)
TLRECL (5)
TLTIME (8)
TLUNIT (3)
TOUTLOC (4)
TOUTSLT (6)
TOUTSRV (4)
TPRDCLN (6)
TPRDINT (6)
TPWTCLN (6)
TPWTINT (6)
TRECFM (4)
TRTNDAT (32)
TSCRIND (1)
TSCRSOR (1)
TTAPLEN (4)
TTAPTYP (2)
TTIMCRT (4)
TTMSCLN (6)
TTRDCLN (6)
TTRDINT (6)
TTRTCH (4)
TTWTCLN (6)
TTWTINT (6)
TUSECLN (6)
TUSECRT (4)
TUSETOT (6)
TUSRDTA (60)
TVENDOR (8)
TVEXDTE (10)
TVEXIND (2)
TVKPDTE (10)
TVKPIND (2)
TVOLCNT (3)
TVOLOWN (8)
TVOLSEQ (5)
TVOLSER (6)

A.5 Field Definitions

This section alphabetically lists and defines all the fields you can use in $RSVP commands.

ABLEN Length of the extent, in tracks.

ABSTR Absolute track address.

ACCOUNT Account code (set by EXIT2 or contained in Format-1 DSCB).

ACTION action set by the ACTION keyword.

ADDRESS Address of the Format-1 DSCB for the data set, in CCHHR format.

ALLOC Data set allocation quantity.

ASSOCC Associated cluster name for a VSAM component. For IPC queries, ASSOCC is blanks for base clusters. For VSAM queries, ASSOCC for base clusters is associated with itself (that is, ASSOCC is the same as the DSNAME).

ASSOCD Associated data component name for a VSAM data set.

ASSOCG Associated alternate index name for a VSAM data set.

#ASSOCG Total number of associated alternate index fields.

ASSOCI Associated index component name for a VSAM data set.

ASSOCR Associated path name for a VSAM data set.

#ASSOCR Total number of associated path name fields.

AVGLRECL Average record length for a VSAM data set.

BILLDAYS Days since last billing.

BILLHRS Hours since last billing.

BLKSZ Physical block size (VSAM - CISIZE).

BUFSIZE Minimum buffer size, in bytes, in virtual storage for a VSAM data set to be provided by the processing program.

C3DSORG Data set organization reported in CA-3 (UCC3) format:

PS = Physical Sequential
PO = Partitioned
VS = VSAM
IS = ISAM
DA = Direct Access
U = Unmovable

CANDVOL Indicates, for a VSAM data set, if the volume is a candidate volume for data and index components. This field contains the value CANDVOL if the volume is a candidate volume, or the value NOCANDV if not. This is a multivalued field.

#CANDVOL Total number of VSAM candidate volume fields.

CASPLITS Number of control area splits for a VSAM data set.

CAT Catalog status:

C = Cataloged
N = Not cataloged
W = Cataloged to another volume
E = Error found in catalog

Specifying CAT causes poor performance. See Performance Notes on page 5-33 in the "Tailoring" chapter for more information.

CB Data set change bit (SU60 compatible). C=SU60 bit on.

CCHH Cylinder and head address.

CDATE DASD file creation date in Julian format (YYYYDDD).

CHCUR Current run charge value (set in $RSEXT1).

CHMONTHS Last twelve monthly charge values.

CHMTD Month-to-date charge value.

CHYTD Year-to-date charge value.

CIPCA Number of control intervals (CI) in a control area (CA) for VSAM data sets.

CISPLITS Number of control interval splits (VSAM components).

COMMENT Comment set by the COMMENT keyword.

CREDT Creation date. Gregorian MM/DD/YYYY

CYLPVOL Cylinders per volume.

CUU Device address.

DATALLOC SMS Data Class average number of bytes required per record.

DATAUNIT SMS Data Class space allocation unit. Values: BYTES, KBYTES, MBYTES.

DATCISZ SMS Data Class VSAM control interval size for KS, ES, or RR.

DATCLASS SMS Data Class name.

DATDATE SMS Data Class last update to date. YYYY/MM/DD

DATDAYSD SMS Data Class calculated field reflecting number of days to retain data set (until deletion) based on data class expiration field. (Zero and negative integer indicate that data set has expired based on data class.)

DATDESC SMS Data Class description.

DATDIBLK SMS Data Class number of directory blocks.

DATE Current date.

DATEXDAT SMS Data Class expiration date.

DATIMBED SMS Data Class VSAM KSDS index option to imbed. Values: IMBED, NOIMBED.

DATKLEN SMS Data Class key length.

DATKOFF SMS Data Class key offset.

DATLRECL SMS Data Class logical record length.

DATPCA SMS Data Class VSAM control area free space percentage.

DATPCI SMS Data Class VSAM control interval free space percentage.

DATRCORG SMS Data Class data set organization. Values: SEQ, KSDS, ESDS, RRDS, LDS.

DATRECFM SMS Data Class record format based on JCL specification. Values: U, V, VS, VB, VBS, F, FS, FB, FBS.

DATREPLC SMS Data Class VSAM KSDS index option to replicate. Values: REPLICATE, NOREPLCA.

DATRETPD SMS Data Class retention period.

DATSHROP SMS Data Class VSAM data set share options.

DATSPPRI SMS Data Class primary allocation of space units.

DATSPSEC SMS Data Class secondary allocation of space units.

DATTIME SMS Data Class time of last update.

DATUSER SMS Data Class ID of last user to update data class.

DATVOLCT SMS Data Class number of volumes for multivolume allocations.

DATXREG SMS Data Class VSAM data set cross-region share options.

DATXSYS SMS Data Class VSAM data set cross-system share options.

DAYSDEL SMS calculated field reflecting number of days to retain data set based on DSCB (non-VSAM) or catalog (VSAM) expiration field. (Zero and negative value indicate that data set has expired based on stored retention at original allocation time.)

DAYSLFTA SMS calculated value reflecting number of days until scheduled archival. (Zero and negative integer indicate available for immediate archival.)

DAYSLFTB SMS calculated value reflecting number of days until scheduled backup. (Zero and negative integer indicate available for immediate backup.)

DAYSLFTD SMS calculated field reflecting number of days until deletion based on management class expired after field. (Zero and negative integer indicate data set has expired based on management class.)

DAYSLFTS Calculated field reflecting number of days eligible to remain in the Disk Staging Area (DSA). (Zero and negative integer indicate data set is eligible for immediate removal.)

DELRECS Number of deleted records (VSAM components).

DEVDB Directory blocks per track.

DEVDT DSCBs per track.

DEVTK Track size.

DEVTYPE UCB device type.

DFGFILE1 Data set name of the first file on a tape created by a $DEFRAG run. $DEFRAG volume records are queried from the IPC if, and only if, a $DEFRAG field is requested. Otherwise, only unload records are returned. If both $DEFRAG type fields and data set type fields are specified on the $RSVP command, both $DEFRAG and data set records are returned. Blanks are returned for data set type fields for $DEFRAG records.

DFGOMED $DEFRAG output media type. Values returned are 3420 (3420 tape device), 3480 (3480 tape device), and M860 (M860 tape device).

$DEFRAG volume records are queried from the IPC if, and only if, a $DEFRAG field is requested. Otherwise, only unload records are returned. If both $DEFRAG type fields and data set type fields are specified on the $RSVP command, both $DEFRAG and data set records are returned. Blanks are returned for data set type fields for $DEFRAG records.

DFGOVOL $DEFRAG run output volume serial number. This is a multivalued field. $DEFRAG volume records are queried from the IPC if, and only if, a $DEFRAG field is requested. Otherwise, only unload records are returned. If both $DEFRAG type fields and data set type fields are specified on the $RSVP command, both $DEFRAG and data set records are returned. Blanks are returned for data set type fields for $DEFRAG records.

#DFGOVOL Total number of volumes used during a $DEFRAG run. $DEFRAG volume records are queried from the IPC if, and only if, a $DEFRAG field is requested. Otherwise, only unload records are returned. If both $DEFRAG type fields and data set type fields are specified on the $RSVP command, both $DEFRAG and data set records are returned. Blanks are returned for data set type fields for $DEFRAG records.

DIRBLKS Number of directory blocks for a PDS unloaded.

DSCOUNT Data set count.

DSIND Data set indicators, in hex.

DSNAME Data set name.

DSORG Data set organization. One of the following values is returned:

PS = Physical Sequential
PO = Partitioned
VS = VSAM
IS = ISAM
DA = Direct Access
U = Unmovable
POE = PDS

DSREC Number of Format-0 DSCBs, which is the number of available DSCBs. For indexed VTOCs, RSVP must be authorized to report the available DSCBs as maintained by MVS.

DSTOT Total number of DSCBs.

DYNARSN CA-Dynam/DASD archive reason. One of the following values is returned:

EXITNOAR	User exit specified do not archive.
ALL	Archive all data type of unload.
IMMED	User requested immediate archive.
NOUSE	Archived due to number of days non-use.
NOTCAT	Archived because it was not cataloged.
EXITARCH	User exit indicated dsn was to be archived.

DYNBRSN CA-Dynam/DASD backup reason. One of the following values is returned:

USERNOBK	User specified no backup.
EXCLUDE	Exclude from backup.
MONTHLY	Backed up during monthly backup.
WEEKLY	Backed up during weekly backup.
IMMED	User requested immediate backup.
USER	User requested backup.
UPDATE	Backed up because data set had been updated.
DAILY	Backed up during daily backup.

DYNDTYPE CA-Dynam/DASD data set indicators. One of the following values is returned:

ICFCAT	Data set is an ICF catalog.
EXPORTED	Data set has been exported.
USERCAT	Data set is a VSAM user catalog.
OVERFLOW	Data set is a track overflow dsn.
ISAM	Data set is an ISAM dsn.
UNMOVE	Data set is unmoveable.
DATASPACE	Data set is a VSAM dataspace.

DYNENQ CA-Dynam/DASD enqueue active at backup time. One of the following values is returned:

EXCL	Data set backed up with exclusive enqueue.
SHARED	Data set backed up with shared enqueue.
NONE	Data set backed up without an enqueue.

DYNRC	CA-Dynam/DASD return code (failure/success indicator).
DYNSYSC	CA-Dynam/DASD system that created the backup tape. Either MVS or DOS is returned.
EDATE	Expiration date in Julian format (YYYYDDD).
ERASE	Indicates records are to be erased (clear to hex zeros) when deleted (VSAM data sets). This field contains the value ERASE if the records are to be erased or the value NOERA if not.
ERROR	Error indications.
ETRACK	Ending track number for a data set.
EXCPEXIT	Exception exit name (VSAM data sets).
EXITFLD	User-defined. The supplied EXIT1 inserts high-level index of the data set name.
EXHICCH	Hexadecimal string indicating the highest extent device address (CC = cylinder, HH = track) of the beginning of the extent (VSAM data sets). This is a multivalued field.
#EXHICCH	Number of highest extent device addresses (VSAM data sets).
EXHIRBA	Highest block relative address of the end of the extent (VSAM data sets). This is a multivalued field.
#EXHIRBA	Number of highest relative block addresses (VSAM data sets).
EXLOCCH	Hexadecimal string indicating the lowest extent device address (CC = cylinder, HH = track). (VSAM data sets). This is a multivalued field.
#EXLOCCH	Number of lowest extent device addresses (VSAM data sets).
EXLORBA	Lowest relative block address of the beginning of the extent (VSAM data sets). This is a multivalued field.
#EXLORBA	Number of lowest relative block addresses (VSAM data sets).
EXPDT	Expiration date in MM/DD/YYYY format.
EXT	Number of extents in the data set.
EXTRKS	Number of tracks in an extent (VSAM data sets). This is a multivalued field.
#EXTRKS	Number of tracks in an extent available (VSAM data sets).
EXTSEQ	Extent sequence number.
FCYLS	Number of full cylinders a data set occupies.
FRCYLS	Number of free cylinders on a volume.
FRTRKS	Number of free tracks on a volume that are not included in FRCYLS. Total tracks = FRTRKS + (FRCYLS * 15).

HIALLRBA Highest allocated relative block address (VSAM data sets). This is a multivalued field.

HIALLRDS Highest allocated relative block address of a VSAM data set.

HIUSERDS Highest used relative block address of a VSAM data set.

#HIALLRB Number of highest allocated relative block address (VSAM data sets).

HIUSERBA Highest used relative block address (VSAM data sets). This is a multivalued field.

#HIUSERB Number of highest used relative block address (VSAM data sets).

IMBED Indicates that the sequence set index record is stored with its associated data control area. (VSAM data sets). This field contains value IMBED if the index record is stored this way, or NOIMB if not.

INDEXLEV Number of index levels used for a VSAM KSDS.

INSRECS Number of records inserted (VSAM data sets).

INUSE Calculated field giving the allocation status of a data set at the time the VTOC for that data set was being analyzed by RSVP.

IPCCOM User comments from the IPC comments cells. You can use the keyword IPCCOML to determine the maximum length reported.

IPCCSTAT Catalog status at unload time of a data set processed by CA-ASM2. These values apply:

CATNONV Non-VSAM, cataloged
NOTCAT Non-VSAM, not cataloged or miscataloged.
VSAM VSAM

IPCDATE Julian date the unload or $DEFRAG processing occurred. The format of the date is YYYYDDD.

IPCDELDT $DA or $DB issued date that this IPC record was deleted. The format of the date is YYYYDDD.

IPCDELID User ID issuing a $DA (Delete Archive) or $DB (Delete Backup) command.

IPCHVOL Home DASD volume for a data set processed by CA-ASM2.

IPCLDEL Indicates the IPC record for the volume or data set has been logically deleted. The field contains the value DELETED to indicate that the record has been logically deleted, or NOTDEL to indicate that the converse is true.

IPCSYSID System ID of catalog environment for dsns processed by CA-ASM2.

IPCTIME Time the CA-ASM2 unload or $DEFRAG was done. The format is HH:MM:SS.

IPCTYPE IPC record type. The values are D for Defrag volume record or U for Unload record.

IPCUTYPE IPC unload record type. The following values apply:

ARCHIVE Archive unload record.
BACKUP Backup unload record.

IPCVTYPE Type of VSAM data set being processed:

C Cluster
S VSAM sphere
G Alternate index
R Path record

When IPCVTYPE is specified in a list or on the command line, RSVP also returns IPC VSAM path records as well as regular normal IPC unload data set records. For path records, IPCVTYPE contains the value R. The DSNAME field is the name of the path. ASSOCC and ASSOCG are also valid for path records. The value of any other non-IPC type fields are unpredictable for path records. Queries that do not access the IPCVTYPE field can only search IPC unload records.

IXRCAND Indicates the data set is a candidate for IXR reload. The single character Y indicates that the data set is a candidate. Otherwise, the field's value is set to the character N.

IXRCDVOL IXR candidate reload volumes. This is a multivalued field.

#IXRCDVO Number of IXR candidate reload volumes.

IXRDEVT Hexadecimal string representing the UCB device type.

IXRFDBK Eight-character hexadecimal string representing the feedback code. Byte 0 is the return code. Byte 1 is the reason code. Bytes 2 and 3 are the DSB information bytes.

IXRLDATE Julian date IXR last modified this IPC record. The date format is YYDDD.

IXRLTIME Time IXR last modified this IPC record. The time format is HH:MM:SS.

IXRRELDA Number of IXR reloads attempted.

IXRRELDC Number of IXR reloads completed.

IXRRLVOL The volume to which IXR reloaded the data set.

IXRSYSID Source system ID of the last IXR update.

IXRTDSN Temporary data set name used by IXR during reload.

KBALLOC Kilobytes allocated.

KBDAYS Kilobyte days since last billing.

KBHRS Kilobyte hours since last billing.

KEYL	Key length.
LASTBKP	Last backup date from the IPC for data sets on disk.
LDATE	Last-used or last-referenced date in Julian (YYYYDDD).
LFCYLS	Largest contiguous extent in cylinders on a volume.
LFTRKS	Largest contiguous extent in tracks on a volume.
LMDATE	Last modification date (MMDDYYYY). This is the only keyword that allows a Gregorian date IF test. Only available to CA-ASM2 sites with the OPEN modification installed.
LMJOB	Last modifying job name, CA-ACF2 logon ID, or account number. Only available to CA-ASM2 sites with the OPEN modification installed.
LMTIM	Time of last modification (HH:MM). Only available to CA-ASM2 sites with the OPEN modification installed.
LPAREN	Open parenthesis (literal).
LRECL	Logical record length (VSAM - maximum record length).
LSTUS	Last referenced date in Gregorian (MMDDYYYY). If you are using this for the IF test, you must use a Julian date. However, it is reported in Gregorian.
L$ARDSN	Defines a literal string: $AR, DA(dsname
L$BKDSN	Defines a literal string: $BK, DA(dsname
L$RLSE	Defines a literal string: RLSE
L$RPVOL	Defines a literal string: VOL(
LV	Last volume. Y if last volume or blank if not.
MAXDAYSD	SMS calculated field reflecting number of days to retain data set based upon maximum retention in the SMS management class. (Zero and negative value indicate that data set exceeds the maximum retention period.)
MDATE	Last modification date in Julian (YYYYDDD).
MGTAUTBK	SMS Management Class automatic backup. AUTOBKP (default), NOAUTOB.
MGTBADU	SMS Management Class allow administrator or user backup command. Values: NONE, ADM, BOTH. Default is BOTH.
MGTBKDY	SMS Management Class number of days to retain backup versions (other than most recent) from the date the backup was created. Range 1 to 9999 or NOLIMIT. Default is 30.
MGTBKFQ	SMS Management Class backup frequency. Range 0 to 9999. Default is 1.

MGTBKNP SMS Management Class number of days to retain the only backup (most recent) version of a deleted data set. Range 1 to 9999, or NOLIMIT. Default is 60.

MGTBKVS SMS Management Class number of backup versions. Range 1 to 13. Default is 2.

MGTBVRD SMS Management Class number of backup versions to keep for a deleted data set. Range 0 to 13. Default is 1.

MGTCLASS SMS Management Class name.

MGTCMAU SMS Management Class automatic migration control. Values: NONE, CMD, BOTH (default).

MGTDATE SMS Management Class last update date. (Gregorian)

MGTDESC SMS Management Class description.

MGTEXDAT SMS Management Class expiration. Values: DATE or NOLIMIT (default).

MGTEXDAY SMS Management Class expire number of days since creation.

MGTEXPDY SMS Management Class expired after number of days of non-use. Values: DATE or NOLIMIT (default).

MGTL1DY SMS Management Class minimum number of days since last access before moving to second level archive. Range 0 to 9999, or NOLIMIT. Default is 60.

MGTMRTDY SMS Management Class maximum number of days to retain. Range 0 to 9999, or NOLIMIT.

MGTPELEM SMS Management Class number of GDGs on primary storage.

MGTPRDY SMS Management Class minimum number of days on primary since last use. Range 0 to 9999. Default is 2.

MGTPREL SMS Management Class data set eligible for partial release. Values: PARTIAL or NOPARTI (default).

MGTRLOMG SMS Management Class for rolled off GDG action. Values: MIGRATE, EXPIRE. No default.

MGTTIME SMS Management Class last update time.

MGTUSER SMS Management Class last update user ID.

MODID Last modifying CA-ACF2 logon ID.

NEWDATCL Value of SMS NEWDATCLASS keyword, padded with blanks to 8 characters.

NEWMGTCL Value of SMS NEWMGTCLASS keyword, padded with blanks to 8 characters.

NEWSTRCL Value of SMS NEWSTRCLASS keyword, padded with blanks to 8 characters.

NUMEXCPS Number of EXCPs performed by VSAM I/O routines (VSAM data sets).

NUMRECS Number of user-supplied records in the data sets (VSAM data sets).

OPTCD Option code, in hex.

ORDERED Indicates volumes are used for allocation in the order they were specified when the cluster was defined (VSAM data sets). The field contains the value ORDERED if the volumes are ordered, or the value NOORDER if not.

OVERFVOL Indicates the volume is an overflow volume on which data records in a key range can be stored (VSAM data sets). The field contains a value of OVERFVOL if the volume is an overflow volume, or the value NOOVERFV if not.

#OVERFVO Number of overflow volume on which data records may be stored (VSAM data sets).

OWNERID Hexadecimal owner identification number (VSAM data sets).

PCA Percentage of free control intervals in a control area (VSAM data sets).

PCI Percentage of free bytes in a control interval (VSAM data sets).

PCT Percentage of space used. For VSAM data sets, PCT is always reported as 100 percent except when a volume type search is being performed and the CLUSTER or DATA keyword is specified.

PHYRSIZE Number of bytes VSAM uses for a physical record in the data or index component (VSAM data sets). This is a multivalued field.

#PHYRSIZ Number of physical record sizes (VSAM data sets).

PHYRTRK Number of physical record sizes (PHYRSIZE) that VSAM can write on a single-track on the volume (VSAM data sets). This is a multivalued field.

#PHYRTRK Total number of physical records/tracks returned (VSAM data sets).

PRIMEVOL Indicates the volume is the first volume on which data records in a key range are stored (VSAM data sets). This field contains the value PRIMEVOL if the volume is a prime volume, or the value NOPRIMEV if not. This is a multivalued field.

#PRIMEVO Total number of prime volume indicators.

PROTECT Indicates if the data set is RACF protected (VSAM data sets). This field contains the value PROTECT if the data set is marked as RACF protected, or the value NOPROTE if not.

PW Password indicators:

Position 1 - Password status: N - None
R - Read and write protection
W - Write protection

Position 2 - RACF status: I - IBM RACF protection

QUOTE Single quote (literal).

RACF Indicates the data set is RACF protected. Values are Y/N.

RACGDG Indicates if the data set is protected by GDG modeling (VSAM data sets). This field contains the value GDGMOD if the data set is protected by GDG modeling, or NOGDGM if not.

RACGPROF Indicates if the data set has a generic RACF profile (VSAM data sets). The field contains the value GENERIC if the data set has such a profile, or NOGENER if not.

RACIND Indicates if the data set is RACF indicated (VSAM data set). This field contains the value RACF if the data set is RACF indicated, or NORACF if not.

RACPNAME RACF profile name (VSAM data sets).

RACPVOL RACF profile volume serial.

RD Rounding used at allocation:

R = Round
N = No round

RECFM Record format:

F = Fixed
V = Variable
U = Undefined
B = Blocked
T = Track overflow
S = Spanned or standard
A = ASA carriage control
M = Machine carriage control

RECOVERY Indicates a temporary close is issued as each control is loaded (VSAM data sets). This field contains the value RECOVERY if the temporary close is issued. Thus, if a serious error occurs during loading, the whole data set does not have to be reloaded. Otherwise, the field contains the value NORECOVE.

RELIND Release indicator:

N Non-enhanced VSAM
E Enhanced VSAM

REPLCATE Indicates index records are to be replicated. This field contains the value REPLCATE if index records are replicated, or the value NOREPLCA if not.

RETPD Number of days to retain data on tape. A value of 9999 sets RETPD to your installation-defined maximum.

RETRECS Number of records retrieved (VSAM data sets).

RKP Relative key position.

RLDALRC Reload routine allocate return code.

RLDCATRC Reload routine recatalog return code.

RLDCOPY Reload copy number of the data set.

RLDFORCE Indicates force was done during reload. The field contains a value of FORCE if force was specified, or NOFOR if not.

RLDIXR Indicates the data was reloaded by IXR. The field contains a value of IXR if the data set was reloaded by IXR, or NOTIXR if not.

RLDLCRC Reload catalog locate return code.

RLDRC Reload return code.

RLDRDATE Date the data set was reloaded. The data format is YYDDD.

RLDRENM Indicates the data set was renamed during the reload. The field contains a value of RENAMED if the data set was renamed, or NORENAM if not.

RLDRTIME Time the data set was reloaded. Time is specified as HH:MM:SS.

RLDRVOL Volume serial to which CA-ASM2 reloaded the data set.

#RLDRVOL Total number of volumes to which CA-ASM2 reloaded the data set.

RLDUCC3 Specifies the data set was reloaded by UCC3 (CA-1) datamover. This field contains a value of UCC3 if the data set was reloaded by this data mover, or NOTUCC if not.

RLDUSER User ID requesting the reload.

RLDVDEVT Hexadecimal string representing the UCB device type field. This is a multivalued field.

#RLDVDEV Total number of reloaded device types.

RLDVIRT Indicates CA-ASM2 reloaded the data set to a virtual volume. The field contains a value of VIRTUAL if the data set was reloaded to a virtual volume, or NOTVIRT if not.

RPAREN Close parenthesis (literal).

SC Specifies the primary source of input as a two-character code. The values returned can be:

BD (BILLDATA)
MS (MSVOLGRP)
VO (VOLUMES)
OC (OSCATALOG)
AC (ASM2CAT)
SV (STGROUP)
DC (DATCLASS)

MC (MGTCLASS)
SC (STRCLASS)
SG (STORGACD)
TM (CA1TMC or TLMSVMF)

SECQ Secondary allocation quantity, in SECT units.

SECT Type of allocation:

A = Absolute track
B = Blocks
T = Tracks
C = Cylinders

SGVCONV SMS controlled volume conversion status.
Values: INCONV, CONVERT.

SGVDATE SMS controlled volume date of last update.

SGVMVSS1 SMS controlled volume MVS system 1 status.
Values: ONLINE, OFFLINE, PENDOFFL, BOXED, NOTREADY.

SGVMVSS2 SMS controlled volume MVS system 2 status.
Values: ONLINE, OFFLINE, PENDOFFL, BOXED, NOTREADY.

SGVMVSS3 SMS controlled volume MVS system 3 status.
Values: ONLINE, OFFLINE, PENDOFFL, BOXED, NOTREADY.

SGVMVSS4 SMS controlled volume MVS system 4 status.
Values: ONLINE, OFFLINE, PENDOFFL, BOXED, NOTREADY.

SGVMVSS5 SMS controlled volume MVS system 5 status.
Values: ONLINE, OFFLINE, PENDOFFL, BOXED, NOTREADY.

SGVMVSS6 SMS controlled volume MVS system 6 status.
Values: ONLINE, OFFLINE, PENDOFFL, BOXED, NOTREADY.

SGVMVSS7 SMS controlled volume MVS system 7 status.
Values: ONLINE, OFFLINE, PENDOFFL, BOXED, NOTREADY.

SGVMVSS8 SMS controlled volume MVS system 8 status.
Values: ONLINE, OFFLINE, PENDOFFL, BOXED, NOTREADY.

SGVSGRP SMS controlled volume Storage Group name.

SGVSMSS1 SMS controlled volume SMS system 1 status.
Values: NONE, ENABLED, QUI/ALL, QUI/NEW, DIS/ALL, DIS/NEW.

SGVSMSS2 SMS controlled volume SMS system 2 status.
Values: NONE, ENABLED, QUI/ALL, QUI/NEW, DIS/ALL, DIS/NEW.

SGVSMSS3 SMS controlled volume SMS system 3 status.
Values: NONE, ENABLED, QUI/ALL, QUI/NEW, DIS/ALL, DIS/NEW.

SGVSMSS4 SMS controlled volume SMS system 4 status.
Values: NONE, ENABLED, QUI/ALL, QUI/NEW, DIS/ALL, DIS/NEW.

SGVSMSS5 SMS controlled volume SMS system 5 status.
Values: NONE, ENABLED, QUI/ALL, QUI/NEW, DIS/ALL, DIS/NEW.

SGVSMSS6 SMS controlled volume SMS system 6 status.
Values: NONE, ENABLED, QUI/ALL, QUI/NEW, DIS/ALL, DIS/NEW.

SGVSMSS7 SMS controlled volume SMS system 7 status.
Values: NONE, ENABLED, QUI/ALL, QUI/NEW, DIS/ALL, DIS/NEW.

SGVSMSS8 SMS controlled volume SMS system 8 status.
Values: NONE, ENABLED, QUI/ALL, QUI/NEW, DIS/ALL, DIS/NEW.

SGVSTAT SMS controlled volume MVS status (includes SGVMVSS1 through 8).

SGVSTGST SMS controlled volume Storage Group status on this volume (includes SGVSMSS1 through 8).
Values: NONE, ENABLED, QUI/ALL, QUI/NEW, DIS/ALL, DIS/NEW.

SGVTIME Time of last update to SMS controlled volume record.

SGVUSER User ID of last user to update SMS controlled volume record.

SHROPTNS Share options (VSAM data sets). The first digit is interpreted as follows:

0 - Data set can be shared by READ users or used by one update user.

1 - Data set can be shared by READ users and used by one update user.

2 - Data set can be fully shared.

3 - Data set can be fully shared with assistance from VSAM.

The second digit describes data set sharing attributes across systems and is interpreted as follows:

2 - Data set can be fully shared.

3 - Data set can be fully shared with assistance from VSAM.

SOURCE Primary input source for the data returned. The values returned can be:

BILLDATA (Billing Input Data)
MSVOLGRP (Mass Store Volume Groups)
VOLUMES

OSCATLOG
ASM2CAT
STGROUPS
DATCLASS
MGTCLASS
STRCLASS
STORGACD
CA1TMC
TLMSVMF

SPACE Blank space (literal).

STGABKP SMS Storage Group automatic backup.
Values: AUTOBKP, NOAUTOB.

STGABSYS SMS Storage Group automatic backup system name.

STGADMP SMS Storage Group automatic dump.
Values: AUTOMIG, NOAUTOM.

STGADSYS SMS Storage Group automatic dump system name.

STGAMIG SMS Storage Group automatic migration.
Values: AUTOMIG, NOAUTOM.

STGAMSYS SMS Storage Group automatic migration system name.

STGDATE SMS Storage Group last update to data.

STGDESC SMS Storage Group description.

STGDMPCL SMS Storage Group dump classes for automatic dump (includes STGDMPC1 through 5).

STGDMPC1 SMS Storage Group automatic dump class 1.

STGDMPC2 SMS Storage Group automatic dump class 2.

STGDMPC3 SMS Storage Group automatic dump class 3.

STGDMPC4 SMS Storage Group automatic dump class 4.

STGDMPC5 SMS Storage Group automatic dump class 5.

STGHTHR SMS Storage Group high threshold. Range 1 to 99 percent.

STGLTHR SMS Storage Group low threshold. Range 1 to 99 percent.

STGPRST SMS Storage Group processor status (includes STGPRST1 through 8).

STGPRST1 SMS Storage Group processor 1 status.

STGPRST2 SMS Storage Group processor 2 status.

STGPRST3 SMS Storage Group processor 3 status.

STGPRST4 SMS Storage Group processor 4 status.

STGPRST5 SMS Storage Group processor 5 status.

STGPRST6 SMS Storage Group processor 6 status.

STGPRST7 SMS Storage Group processor 7 status.

STGPRST8 SMS Storage Group processor 8 status.

STGROUP SMS Storage Group name.

STGSTAT SMS Storage Group status.
Values: NONE, ENABLED, QUI/ALL, QUI/NEW, DIS/ALL, DIS/NEW.

STGTHRSH SMS Storage Group threshold status.
Values: THRESHOLD, NOTHRESHO.

STGTIME Time of last update to SMS Storage Group.

STGTYPE SMS Storage Group type.
Values: POOL, VIO, DUMMY.

STGUSER User ID of last user to update SMS Storage Group.

STGVMAX SMS Storage Group VIO maximum data set size.
Range: 8 to 2,000,000 kbytes.

STGVUNIT SMS Storage Group generic device type to be simulated by VIO.

STORGACD SMS database search.

STRACK Starting track number on which a data set resides.

STRAVAIL SMS Storage Class data availability specification. Values: NONE, STD, CONT.

STRCLASS SMS Storage Class name.

STRDATE Date of last update to SMS Storage Class.

STRDBIAS SMS Storage Class direct Bias.
Values: NONE, READ, WRITE.

STRDESC SMS Storage Class description.

STRDRESP SMS Storage Class direct millisecond response.
Range: 1 to 999.

STRGSP SMS Storage Class guaranteed space.
Values: GUARANTEE, NOGUARANT.

STRSBIAS SMS Storage Class sequential bias.
Values: NONE, READ, WRITE

STRSRESP SMS Storage Class sequential millisecond response.
Range: 1 to 999.

STRTIME SMS Storage Class last update time.

STRUSER SMS Storage Class last update user ID.

SUL Indicates if standard user labels are associated with the data set.
Values are Y or N.

TACTFIL	Tape active files on volume (TLMS).
TAUDDTE	Tape last audit date (CA-1).
TAUDTIM	Tape last audit time (CA-1).
TBADIND	Tape bad tape indicator (CA-1). Values are Y or N.
TBASVOL	Tape first volume, mother volume (CA-1/TLMS).
TBLKCNT	Tape block count (CA-1/TLMS).
TBLKSIZ	Tape block size (CA-1/TLMS).
TCDATE	Tape creation date (CA-1/TLMS).
TCDDNME	Tape creation ddname (CA-1/TLMS).
TCDSSEQ	Tape CDS file sequence (TLMS).
TCJOB	Tape creation job (CA-1/TLMS).
TCLNIND	Tape clean indicator (CA-1). Values are Y or N.
TCLSIND	Tape close indicator (CA-1/TLMS). Values are O, C or A. O indicates open, C indicates closed, and A indicates closed by abend.
TCPGM	Tape creation program name (TLMS).
TCPUID	Tape CPUID (CA-1/TLMS).
TCSTEP	Tape creation step (CA-1/TLMS).
TCTIME	Tape creation time (CA-1/TLMS).
TCUNIT	Tape creation unit (CA-1/TLMS).
TDATCRT	Tape date certified (TLMS).
TDATSCR	Tape date scratched (TLMS).
TDELIND	Tape delete indicator (CA-1). Values are Y or N.
TDEN	Tape density (CA-1/TLMS).
TDKPDTE	Tape data set keep date (TLMS).
TDKPTYP	Tape data set keep type (TLMS).
TDSNACT	Tape active DSNB indicator (CA-1).
TDTECLN	Tape date cleaned (CA-1/TLMS).
TDTEMOV	Tape date moved (CA-1/TLMS).
TDYNIND	Tape DYNAM/T controlled indicator (CA-1). Values are Y or N.
TEDMIND	Tape EDM controlled indicator (CA-1). Values are Y or N.
TEXPDT	Tape expiration date (CA-1/TLMS).
TEXPDTP	Tape expiration type date (CA-1 5.1 only/TLMS).

TFILCNT	Tape file count (CA-1/TLMS).
TFILSEQ	Tape file sequence number (CA-1/TLMS).
TFSTVOL	Tape first volume (CA-1).
THD1DSN	Tape DSN17 (CA-1).
TIME	Current time HH:MM:SS.
TINPUT	Tape input process (CA-1. Only available when used with 5.1)
TINTCIN	Tape internal change indicator (CA-1). Values are Y or N.
TINTDTE	Tape initial date (CA-1/TLMS).
TLABTYP	Tape label type (CA-1/TLMS).
TLDATE	Tape last use date (CA-1/TLMS).
TLJOB	Tape last use job name (CA-1/TLMS).
TLRECL	Tape logical record length (CA-1/TLMS).
TLTIME	Tape last use time (CA-1/TLMS).
TLUNIT	Tape last use unit address (CA-1/TLMS).
TLUPDAT	Tape last update cause (CA-1).
TMANUPD	Tape manual update indicator (CA-1. Only available when used with 5.1.)
TNXTVOL	Tape next volume (CA-1).
TOOUTIN	Tape opened for output indicator (CA-1). Values are Y or N.
TOUTLOC	Tape output location code (CA-1/TLMS).
TOUTPUT	Tape output process (CA-1. Only available when used with 5.1)
TOUTSLT	Tape out slot (CA-1/TLMS).
TOUTSRV	Tape out-of-service indicator (CA-1/TLMS).
TPRDCLN	Tape permanent read clean (CA-1/TLMS).
TPRDINT	Tape permanent read initialize (CA-1/TLMS).
TPREVOL	Tape previous volume (CA-1).
TPWTCLN	Tape permanent write clean (CA-1/TLMS).
TPWTINT	Tape permanent write initialize (CA-1/TLMS).
TRDCAT	Tape read by catalog indicator (CA-1). Values are Y or N.
TRDSOVR	Tape RDS override indicator (CA-1). Values are Y or N.
TRECFM	Tape record format (CA-1/TLMS).
TRECIND	Tape re-created indicator (CA-1). Values are Y or N.
TRKALLOC	Tracks allocated.

TRKDAYS Track days since last billing.

TRKHRS Track hours since last billing.

TRKPCYL Tracks per cylinder.

TRKPVOL Number of tracks on a DASD volume.

TRKSCA Number of tracks per control area within an extent (VSAM data set). This is a multivalued field.

TRTNDAT Tape retention data (TLMS).

TSCRFLG Tape scratch flag (CA-1).

TSCRIND Tape common scratch indicator (CA-1/TLMS). Values are Y or N.

TSCRSOR Tape scratch source (TLMS).

TTAPLEN Tape length (TLMS).

TTAPTYP Tape type (TLMS).

TTIMCRT Tape times certify (TLMS).

TTMSCLN Tape times cleaned (CA-1/TLMS).

TTPMIND Tape temporary data set indicator (CA-1). Values are Y or N.

TTRDCLN Tape temporary read clean (CA-1/TLMS).

TTRDINT Tape temporary read initialize (CA-1/TLMS).

TTRTCH Tape recording technique (CA-1/TLMS).

TTWTCLN Tape temporary write clean (CA-1/TLMS).

TTWTINT Tape temporary write initialize (CA-1/TLMS).

TUSECLN Tape uses cleaned (CA-1/TLMS).

TUSECRT Tape uses certify (TLMS).

TUSERID Tape user ID (CA-1).

TUSETOT Tape total uses (CA-1/TLMS).

TUSRDTA Tape user data (CA-1/TLMS).

TUSRIND Tape updated user indicator (CA-1). Values are Y or N.

#TRKSCA Total number of tracks per control area within an extent (VSAM data set).

TV Type of VSAM data set:

- **K** Key-Sequenced Data Set (KSDS)
- **E** Entry-Sequenced Data Set (ESDS)
- **R** Relative Record Data Set (RRDS)
- **L** Linear Data Set (Linear)

TVENDOR Tape vendor (CA-1/TLMS).

TVEXDTE Tape volume expiration date (TLMS).

TVEXIND Tape volume expiration type (CA-1. Only available when used with 5.1.)

TVKPDTE Tape volume keep date (TLMS).

TVKPIND Tape volume keep type (CA-1. Only available when used with 5.1.)

TVOLCNT Tape volume count (TLMS).

TVOLOWN Tape volume owner - EDM ID (CA-1/TLMS).

TVOLSEQ Tape volume sequence (CA-1/TLMS).

TVOLSER Tape volume serial number (CA-1/TLMS).

TYPE Supplied $RSEXT1 exit inserts TEMP for temporary data sets. Supplied $RSEXT11 exit inserts EMTY for empty nontemporary data sets. This field cannot be interrogated using the IF/AND keywords. To select empty data sets see the discussion of $RSEXT11 and $RSEXT12.

TXBYCAT Tape expired by catalog (CA-1).

TXBYCYC Tape expired by cycle (CA-1).

TXBYJUL Tape expired by Julian (CA-1).

TXBYLDT Tape expired by last date (CA-1).

TXPROC Tape exception process (CA-1. Only available when used with 5.1.)

UC3DEN UCC3 tape density at unload time. One of the following values is returned: 38K, 6250, 1600, 800.

UC3DSEQ UCC3 disk volume file sequence number at unload time.

UC3FEET UCC3 calculated number of feet the data set occupies on tape. For 3480 cartridges, this value is actually in inches.

UC3FSEQ UCC3 logical tape file sequence number at unload time.

UC3RUN UCC3 run number that processed this data set.

UC3VSAM UCC3 VSAM data set type:

NONE
KSDS
ESDS
RRDS

UC6PSWD UCC6 partitioned data set.

UC6ST UCC6 partitioned data set status bytes. This field is a hexadecimal string representing the two UCC6 status bytes.

UNIKEYS Indicates non unique keys are allowed (VSAM data sets). This field contains a value of UNIKEYS if the alternate key value identifies one, and only one, data record in the base cluster. Otherwise, this field is reported as NOUNIKRE.

UNIQUE Indicates that the data set or index must reside in a data space all its own. The field contains the value UNIQUE if the data must be in its own data space, or the value NOUNIQ if not. (VSAM data set.)

UNIT Unit name.

UNLAERR Indicates an access error occurred for this unload version. The field contains the value ERROR if an access error occurred, or NOERR if not.

UNLARCAT Indicates a recatalog was attempted for the data set during unload. The field contains the value ATRCAT if a recatalog was attempted, or NOTDELxx if not.

UNLASCRT Indicates a scratch was attempted for the data set during unload. The field contains the value ATSCRT if a scratch was attempted, or NOTDELxx if not.

UNLAUCAT Indicates an uncatalog was attempted for the data set during unload. The field contains the value ATUCAT if an uncatalog was attempted, or NOATUC if not.

UNLBKBV Number of kilobytes unloaded on the volume.

UNLBLK Unloaded volume starting block number. This field is displayed in hexadecimal.

UNLBLKSZ Physical block size of the unloaded data set on the output media.

UNLBYTES Actual number of bytes unloaded.

UNLBYTKB Bytes unloaded in Kilobytes.

UNLCBYTE Data set byte count on the output media after compression.

UNLCKBC Data set byte count on the output media after compression in kilobytes.

UNLCOMP Indicates if the unloaded data set is compressed. The field contains the value COMPRES if the data set is compressed, or NOTCOMP if not.

UNLCOPY Number of unloaded copies.

UNLCTECH The data compression technique used to unload the data set. Values returned:

NOTCOMP
STD
FAST
TEXT

UNLEMPTY Indicates if the unloaded data set was empty at unload. The field contains the value EMPTY if the data set was empty, or NOEMP if not.

UNLFORM Character string representing the unloaded data's format. The following values are returned:

$PDM	CA-ASM2 physical data mover format
$PDS	(modified) IEHMOVE format
IEBGENER	Standard IBM IEBGENER format
IEBCOPY	Standard IBM IEBCOPY format
IDCAMS	Standard IBM IDCAMS format
IEBISAM	Standard IBM IEBISAM format
FDR	FDR utility format
SAS	SAS output format
$ULBDAM	CA-ASM2 direct-access data set format
UCC6	IEBCOPY requiring reformat after reload

UNLFSEQ File sequence number of the unload volume.

UNLFVOL Indicates if the data set was unloaded during a full-volume backup. The field contains the value FULLVOL if the backup was full volume, or NOTFULL if not.

UNLIBKP Indicates if the data set was unloaded during an incremental backup. The field contains the value IBKUP if the backup was incremental, or NOTIB if not.

UNLMAMT Total number of bytes occupied by the data set in the Disk Staging Area (DSA). Total represents data bytes after compression and overhead.

UNLMDTYP Character string representing the unload media type. Values returned:

3420 (3420 tape device)
3480 (3480 tape device)
M860 (M860 tape device)
348X (IDRC tape device)
3330 (3330 disk device)
3340 (3340 disk device)
3350 (3350 disk device)
3375 (3375 disk device)
3311 (3311 disk device)
3380 (3380 disk device)
3390 (3390 disk device)
6421 (Fujitsu disk device)
UNKN (Unknown disk device)

UNLMGR Character string representing the disk manager for this unload. The current values are ASM2 and UCC3.

UNLNRBLK Indicates if the data set was not reblocked at unload time. The field contains the value NOTREBL if the data set was not reblocked, or REBLOCK if it was.

UNLODSN Name of the data set on the unload media.

UNLPCYCL Indicates if the data set was unloaded due to inactivity to the value of $CYCLETM. The field contains the value CYCLE if the data set was unloaded due to inactivity, or NOCYC if not.

UNLPERM Indicates the data set was unloaded to a permanent tape. The field contains the value PERMTAP if a permanent tape was desired, or NOTPERM if not.

UNLRCAT Indicates if a recatalog was successful for the data set. The field contains the value RCATOK if the recatalog was successful, or NORCAT if not.

UNLRCATP Indicates if the data set was recataloged to a pseudo volser at unload. The field contains the value PSEUDORC if the recatalog was to a pseudo volser, or NOPSEUDO if not.

UNLRETPD Retention period, in days, specified for the unloaded data set.

UNLSCRT Indicates if the data set was scratched at unload. The field contains the value SCRTOK if the data set was scratched, or NOSCRT if not.

UNLSYSID System ID where the unload was performed.

UNLTRKS Number of tracks occupied by the unloaded data set in the Disk Staging Area (DSA).

UNLUARC Indicates if the data set was unloaded due to a user-initiated archive. The field contains the value UARCH if the archive was user initiated, or NOTUA if not.

UNLUBKP Indicates the data set was unloaded due to a user-initiated backup. The field contains the value UBKUP if the backup was user initiated, or NOTUB if not.

UNLUCAT Indicates if the data set was uncataloged at unload. The field contains the value UCATOK if the uncatalog was successful, or NOUCAT if not.

UNLVERS Character string representing the major version of CA-ASM2 used for this unload. Versions are represented as 400, 300, 250, 240, 230, and 220. As an example, the string 300 would represent Version 3.1, 3.0.3, 3.0.2, or 3.0.1.

UNLVIRT Indicates if the data set was unloaded from a virtual volume. The field contains the value VIRTUAL if the data set was unloaded from a virtual volume, or NOTVIRT if not.

UNLVOL Volume serial of the tape or disk drive to which the data set was unloaded.

UNLV# Unload data set version number in the IPC.

UNUSED Quantity of unused space.

UPDRECS Number of updated records (VSAM data sets).

USECNT Data set use count. Only available to CA-ASM2 sites with the OPEN modification installed.

USED Quantity of space used.

VEXTS Number of extents allocated in this set of extents on this volume (VSAM data set). This is a multivalued field.

#VEXTS Total number of VEXTS fields available.

VOLALLOC Amount of allocated space on the volume.

VOLATTR Volume attributes (PUBLIC, PRIVATE, or STORAGE).

VOLCAP Volume capacity in megabytes.

VOLEXT Count of free space extents.

VOLLSPC Text from the IBM macro LSPACE SVC.

VOLNDX Volume index VTOC status:

NOTVAL Index is not valid.
ENABLE A valid index is active.
DISABL A valid index is not active.

VOLPOOL User-defined. Supplied EXIT1 inserts data from &V control statements.

VOLSPC Quantity of free space on volume.

VOLSQ Volume sequence number.

VOLUME Volume serial number.

VOL1XSP Largest single extent of free space.

VPCT Percentage of allocated space on the volume.

VS VSAM entity type:

M = Master catalog
U = User catalog
C = Cluster
I = Index
G = Alternate index
D = Data
P = Page space
S = Swap space

Note: If DATA is specified, VS contains either I or D. If CLUSTER is specified, VS does not contain either I or D. If DATASPACE is specified, VS is always blank.

VTOCE VTOC extents, in CCHH format.

VTOCI VTOC indicators, in hex.

VTOCL VTOC length, in tracks.

VXTYPE Extent type (VSAM data set):

00 Extents are contiguous
40 Extents are not preformatted
80 Sequence set occupies a track adjacent to a control area
CO VSAM data set converted from a VSAM to an ICF catalog
FF Candidate volume

This is a multivalued field.

#VXTYPE Total number of VXTYPE fields available.

WRTCHK Indicates if write operations are to be checked for correctness. This field contains the value WRTCHK if write operations are to be checked, or the value NOWRTC if not.

WEEKDAY Current weekday name.

YDD Current date in Julian format (YYDDD).

Glossary

Aging. Aging is the process of maintaining data set usage or activity information. Aging information is critical to many CA-ASM2 functions, and can be maintained by CA-ASM2 code, the client's own code, or mainstream IBM code (MVS SU60, or the IBM program product DF/DS for VSAM). All CA-ASM2 functions operate without restriction, regardless of the aging method chosen.

Allocation Manager. Allocation Manager is a utility that enforces standards for the allocation of new DASD data sets and restricts access to selected data sets, volumes, and units. It controls the allocation of new data sets by overriding the UNIT parameter based on information center-defined criteria. It restricts access to selected DASD resources by checking new DASD allocations and references to existing DASD resources against a table of restricted data specified by your information center. Allocation Manager does not handle VSAM data sets.

Archive. Archive involves moving data from online DASD to less expensive media, either tape or disk. Data sets are normally eligible for archive when not accessed on DASD for a period of time. Archived data is not directly accessible by user applications, except when it has been recataloged to the tape volume, or when Intelligent Transparent Restore (IXR), which automatically reloads requested data sets, is implemented.

Archive (noun) is the storage medium, tape or disk, containing data sets that were copied, then scratched from DASD.

Archive (verb) is to move data from DASD to an archive medium.

Archived (adjective) describes a data set that has been copied to an archive medium.

Archive/Backup medium. Refers to either tape or disk. Data sets can be archived/backed up to tape or to a secondary disk known as a Disk Staging area (DSA). Normally, disks in the DSA are slower (and cheaper) than primary online DASD.

ARCPARMS. ARCPARMS is a data set (member of CAI.PARMLIB) that contains parameters that apply to any type of archive or backup run, and establish standard criteria for archive and backup runs. The supplied procedures for archive and backup runs point to this data set.

Automatic purge. When the disk-to-disk unload mode is used for any archive or backup by $DASDMNT, if an allocation into the Disk Staging Area (DSA) fails for lack of space, $DASDMNT can perform an automatic purge of the unload copies in the DSA.

Backup. Backup protects data from corruption by providing an exact copy, which can be used to re-create the original data should it become corrupted or destroyed. A backup copy may be an individual data set or a full volume. The backup copy is made to a backup medium, tape or disk, while the original version is still available on online DASD. The backup copy is not directly accessible by a user application, and must be restored to online DASD before it can be accessed. Backup (noun) is the storage medium, tape or disk, containing duplicate copies of data sets residing on DASD volumes.

Back up (verb) is to copy the data from DASD to the backup medium.

Backup (adjective) describes a data set that has been copied to a backup medium.

Batch retrieval. Batch retrieval involves submitting $RA or $RB commands as in online retrieval, but the commands are input to a batch retrieval run. The ASM2CMDU procedure is the batch command processor which is then used to process the retrieval requests.

CA-1. A full function tape management and vault management system for OS systems. It was formerly known as UCC1 or TMS.

CA-RSVP. Report Selection and Variable Processing. A flexible utility product that is distributed as a

component of CA-ASM2. It will read VTOCs, ICF catalogs, the CA-ASM2 IPC, CA-1 TMC, CA-DYNAM/TLMS VMF, or any user-defined data set, select a subset of those records as described by the user's selection criteria, and produce reports or extract files containing user-selected data fields.

Controlled Scratch. The Controlled Scratch component automatically removes data sets based on information center-defined criteria when those data sets are no longer needed. Once criteria for controlled scratching have been established, CA-ASM2 determines which data sets are to be scratched and automatically scratches them at the designated time.

Controlled Scratch offers two methods of automatically scratching data sets using predetermined criteria; by specifying criteria for the selection of data sets to be scratched or specifying criteria to identify data sets to be protected and scratching all others.

Data Set Placement (DSP). $DEFRAG can be directed to place data sets in a predetermined position on the volume(s) being defragmented. Controlling the positioning of data set placement on volumes can greatly optimize DASD performance.

Improper data set placement can intensify the impact of head contention and valuable processing time is spent traversing the disk between concurrently used files rather than performing file I/O operations. Head contention can lead to erratic patterns in online response time and batch turnaround time. This causes overall system performance to be unpredictable and less than optimum. Proper use of the DSP option can eliminate serious contention problems and reduce the total seek time needed, contributing to significantly improved system performance.

Deferred Scratch. The Deferred Scratch option defers all scratch and uncatalog operations of archived data sets until CA-ASM2 has created a duplicate of the tape on which the data sets reside. This option requires the appropriate bit to be set in the $MISCOPT field in $OPTIONS. If this option is not selected, the scratch and uncatalog take place immediately after the data move to tape and before the duplex step.

Deferred Scratch provides additional integrity for the archived data; however, the deletion of data sets could be considerably delayed if the duplex step, for whatever reason, fails to complete.

Deferred Scratch is an option for all disk-to-tape archival, it is not supported for disk-to-disk archival.

Defragmentation. Volume defragmentation (also referred to as volume compression) solves the common problem of the fragmentation of unallocated space on a volume. A fragmented volume results when data sets have extended into multiple, noncontiguous extents, and after numerous deletions of data sets leave gaps of unallocated space across the volume. Volume defragmentation is the process by which all free space extents are consolidated, and movable data sets rearranged, to produce a volume with the largest possible contiguous free space areas.

Severe fragmentation can seriously affect system performance by diminishing I/O access speed and can result in poor utilization of expensive DASD space.

Differential Backup. Refers to an incremental backup of data sets based on the SU60 bit in the Format-1 DSCB and NOT resetting the SU60 bit after a successful backup. Thus, each differential backup contains all the data that has changed since the last full-volume backup. This type of incremental backup can be useful when extremely quick turnaround is required for incrementally recovering a pack.

Disk-to-Disk. The disk-to-disk feature provides the option to archive and back up data sets to disk rather than to tape. The disk archive or disk backup can be thought of as a Disk Staging Area (DSA) where CA-ASM2 holds the archive or backup data sets until they can be copied to tape. Data sets archived or backed up to disk may be directly retrieved or restored, if required.

Disk Staging Area (DSA). This is a disk pool identified by an esoteric unit name or disk SMS storage group that holds user data sets which CA-ASM2 has archived or backed up. Data sets in the DSA should be copied to tape so that multiple copies exist. CA-ASM2 can reload data sets from the DSA more quickly, but data sets do not reside in the DSA for the remainder of their life cycle. As space in the DSA becomes short, older data sets in the DSA are purged to make room for newer ones.

Disk Staging Area Manager (M2DSAMGR). The DSA Manager program manages the data sets within the DSA. It performs the following functions; copies data sets from the DSA to tape, purges data sets from the DSA, and lists the contents of the DSA.

DDSC. Disk Destination Subcell. This is the subcell within the UDC that specifically identifies an archived or backed up data set in the DSA.

DSA. Disk Staging Area. See description above.

DVR. DEFRAG Volume Record. Contains information such as data and time of the full-volume dump, output media type, and the tape volume serials used.

Explicit Archive. This refers to an archive that is run at a user's request (also called user-initiated).

Explicit Backup. This refers to a backup that is made at a user's request (also called user-initiated).

Forward Merge. Forward merge is a CA-ASM2 process that consolidates unexpired archived or backed up data sets. Data sets on existing tapes are consolidated on a new tape and expired data sets encountered on tape are dropped.

Full-volume dump/restore. Refers to dumping an entire DASD volume. $DASDMNT can dump an entire DASD volume to tape and then restore it in its original format. The Stand-Alone Restore (SAR) facility uses the full-volume dumps made by $DEFRAG.

Group Name Modification. This portion of the Allocation Manager facility defines the DASD pools by allowing the user to dynamically reconfigure esoteric unit names defined to the system. Esoteric unit name groups are managed by DASD volume serial numbers rather than unit addresses. Processing is initiated by executing a batch job at IPL time or when the volume configuration changes.

Incremental Backup. The most common system-initiated backup is known as incremental backup. Incremental backup automatically backs up data sets that have been modified since the previous backup. This permits a more nearly automatic restore if a volume requires recovery. Optionally, all data sets modified since the last full volume backup can be backed up.

Incremental Recovery. Incremental recovery restores an otherwise unusable DASD volume to its state when it was last backed up incrementally. This implies that CA-ASM2 is incrementally backing up modified data on a volume to complement periodic full-volume backup.

Integrated Product Catalog (IPC). The CA-ASM2 Integrated Product Catalog which contains information on every data set backed up or archived by CA-ASM2. It also contains information on all full-volume dumps created by the CA-ASM2 full-volume dump utility, $DEFRAG.

Many fields of data, including Format-1, data class, and SMS data, are stored in the IPC. The IPC is a direct-access catalog, allowing for information to be obtained quickly. It has an ISPF interface for quick inquiry and update capabilities.

Intelligent Transparent Restore (IXR). The Intelligent Transparent Restore facility automatically and transparently restores archived data sets accessed by a batch job, time-sharing user, or started task. No manual process is required of the user. With IXR your information center can even archive production data sets since IXR will automatically restore them when production jobs are scheduled.

Journal. The journal data set is basically a backup copy of the IPC. The IPC integrity scheme is based on saving copies of the IPC at various points in time and on recording update activity in a separate journal data set. (The journal should be on a separate DASD volume from the IPC.) When CA-ASM2 has successfully backed up the IPC, it clears the journal data set and writes a special record into the journal which holds the data set name and volume information of the backup volume(s) just created. The Journal is then available to record new activity until the next IPC backup cycle. If the IPC is destroyed, the journal contains all the information needed to complete a recovery.

Library Compare Utility. This utility compares members of multiple libraries and reports all duplicate members through a report. This report provides a listing of all members found on more than one library with the assembled/compiled date and the link edit date for each load module or the ISPF last modified and creation dates for each source member for which ISPF statistics exist. You can use the information provided by the listing to eliminate duplicate members on various libraries if it is not necessary for them to be there.

Logical Data Mover. A logical data mover moves data from one location to another using a logical record or block as the unit of data transfer. The term logical is used because the data mover requires that the data set being moved conforms to the logical format (its self-describing characteristics) in the Format-1 DSCB. Logical inconsistencies often result in unusable data, and the logical data mover will generally not process unusable data.

Migrate. Migration refers to moving data sets from one device to another, either tape or disk, while maintaining immediate access to the data. The migration facility can migrate both sequential and partitioned nonkeyed data sets to disk or MSS, but only sequential data sets to tape. It can reblock data sets to new block sizes that are compatible with the target volumes. Data sets may be migrated freely between SMS and non-SMS controlled

volumes. When moving data sets to SMS controlled volumes, separate SMS classes may be assigned to each data set moved, but your information center's ACS routines may override any or all class specifications.

The migration facility is not limited to the $MIGRATE program, but is supported by the archive process as well.

Online retrieval. Online retrieval provides immediate reload of archived data sets, and immediate recovery of backed up data sets. This is not recommended if operational efficiency is a major concern. It is recommended if convenience to the user is the overriding objective.

Partial-volume dump/restore. Refers to dumping and restoring non-VSAM and ICF VSAM data sets on an individual or prefix basis. $DEFRAG can do both and also restore parts of data sets on a relative track within a data set or CCHH basis, if the data sets have not changed location on the volume.

Physical Data Mover. The high-speed physical data mover copies full cylinders of data and conserves valuable tape since it does not copy unused tracks. This type of data mover is faster than the logical data mover but does not permit device-independent restores.

Preprocessing. Preprocessing refers to any preliminary processing step that sets the stage for the actual processing to be performed. CA-ASM2 features two preprocessing facilities, the Report Selection and Variable Processing (CA-RSVP) facility, and GDGMON. CA-RSVP scans the Volume Table of Contents (VTOCs) of specified volumes to select candidates for further processing, such as archive, backup, scratch, compression, or reporting. GDGMON functions similarly to CA-RSVP, but it preprocesses GDG data sets whose relative generation numbers exceed a given value.

Protect data set. The protect data set allows you to register data sets that are never to be scratched by Controlled Scratch. If you find certain data sets repeatedly being selected for scratch and withheld from the actual scratch process, you may want to include them in the Protect data set. The Protect data set can be used to protect specific data sets. This includes all occurrences of the data set or occurrences only on specified volumes.

Queue. When a request to unload or reload a data set is submitted, the data set name is placed in a queue (a physical sequential data set) to await further processing. The various queues (archive, backup, reload, and restore) are serviced on a regular basis, usually once a day, at which time the unloads or reloads are performed.

Queue Manager. The Queue manager facility manages the CA-ASM2 command queue. The $QM command lists or deletes queued archive or backup command entries. You select queued command entries to be listed or deleted by identifying various fields or combinations of fields. An alternative to submitting a $QM command is to use the ISPF Queue Manager panel.

Queued retrieval. This form of retrieval may use online facilities for the entry of a reload/restore request, but performs the actual data reload/restore in a batch run. The ASM2CMDU procedure is used to perform the batch run.

Realtime Space Monitor (RTSM). The Realtime Space Monitor performs space reporting and disk-to-disk threshold archiving. RTSM actively monitors SMS storage groups, unit groups, and volumes for space availability and utilization. It captures space information on a time interval basis and records this information in a data set for later processing. As it records the free space on the monitored volumes, it determines if the storage utilization exceeds threshold values defined by your information center. Threshold archival begins when RTSM detects the high threshold value and continues until enough data has been moved to attain the low threshold value.

Reload. Reload is the process of transferring data from an archive medium to an online DASD volume. The term reload is sometimes used generically to denote the process of moving data from either an archive or backup medium to an online DASD volume.

Restore. Restore is the process of transferring data from a backup medium to an online DASD volume. A restore-from-backup is usually performed when the original online copy has been corrupted.

Retrieval. This is the process whereby data sets that have been archived or backed up to tape or disk are returned to online DASD. CA-ASM2 provides three ways of retrieving archived and backed up data; online retrieval, batch retrieval, and queued retrieval. Another method of retrieving data is through the Intelligent Transparent Restore (IXR) facility which provides automatic realtime retrieval of data sets.

Simulated mode. Live mode is the normal mode of operation and is assumed unless the $SIMUL SYSIN parameter is present. In simulated mode, the $DASDMNT program reports recommended actions for data sets, but does not execute the operation. Simulated

mode is especially useful when tuning CA-ASM2 system parameters or testing your information center exits.

SMS. 1. System Managed Storage.

A philosophy of storage management that has the system management software making storage management decisions.

2. Storage Management Subsystem.

Also referred to as DFSMS, a component of IBM's DFP 3.1 and above system that provides a mechanism for associating Storage, Management, and Data classifications with individual data sets and also provides a means of defining disk pools called Storage Groups. CA-ASM2 interfaces with SMS to automatically take care of data set availability.

Stand-Alone Restore (SAR). The Stand-Alone Restore (SAR) facility offers reliable recovery from system crashes by recapturing the system volume in a stand-alone environment. SAR is composed of load modules that form a mini operating system. A separate program invokes these modules to produce an SAR load tape or disk, either of which can be used to restore critical volumes for IPL of the system. The SAR load modules can be stored on disk and invoked directly from disk rather than tape, which yields a faster, more efficient restore.

System-Initiated archival. Archival increases online disk space availability by moving infrequently used data from disk-to-tape or from disk-to-disk. System-initiated archival (also called system-pass archive) automatically archives data to tape or disk based on criteria defined by your information center.

TDSC. Tape Destination Subcell. This is the subcell within the UDC that specifically identifies an archived or backed up data set. If multiple copies of the data set have been made, there will be one TDSC for each duplex.

Threshold Archival. Threshold archival provides maximum disk availability by archiving data sets based on threshold values defined by your information center. It is available only as disk-to-disk archival with the Realtime Space Monitor, but either disk-to-tape or disk-to-disk with the $TA command.

The Threshold Archiver analyzes a specific volume or group of volumes to determine if the amount of available space satisfies installation defined threshold levels. If sufficient space is available, the threshold archiver simply terminates. When the threshold is exceeded, it queues data sets for archival until the threshold is satisfied. The SMS storage group low threshold value or command input is used as the threshold value. When archiving is required, data sets are selected based on inactivity (least most recently referenced) and data set size.

UDC. Unload Destination Cell. An IPC record that identifies a data set that has been backed up or archived has a UDC. The UDC contains information related to the whereabouts of that unloaded data set and all of its duplex copies.

Unload. Unload refers to the act of archiving or backing up a data set to either tape or disk. Archive and backup (unload) operations may be activated by demand or scheduled.

$COPY. The $COPY utility allows you to copy input files of any RECFM to an output print file with the RECFM or FA, FBA, VA, or VBA. If the RECFM of the input files specifies ASA control characters, they will be maintained in the output file. If ASA is not specified, the first position of the output file is set to a space.

$COPYTP. $COPYTP is a tape copy utility that creates copies of CA-ASM2 archive and backup tapes. These tapes provide backup in case a tape is lost or becomes unreadable.

$DASDMNT. This DASD maintenance program is executed for archive or backup functions. It oversees the unload process, controlling the data sets selected for archive or backup based on explicit requests or general parameters. SYSIN parameters define the mode of operation of the $DASDMNT run and establish criteria for the archive or backup run.

$DEFRAG. The $DEFRAG utility provides full-volume or partial-volume dump, full-volume restore, selective data set restore from a full-volume dump, full-volume defragmentation, and Data Set Placement. $DEFRAG can defragment (compress) DASD volumes without using a disk-to-disk copy function. Together with incremental backup, $DEFRAG provides your information center with a complete system for backing up and restoring DASD data.

$GDGMON. $GDGMON functions as a preprocessor. It is designed specifically to generate archive, backup, or migrate commands for data sets contained in generation data set groups. By inspecting VTOCs of selected volumes, $GDGMON can generate requests to archive, backup, or migrate generation data sets whose relative generation number exceeds a given value. Optionally, it can scratch, uncatalog, or list selected generation data sets. SYSIN parameters allow separate actions for miscataloged and uncataloged GDG data sets.

$MAINT. $MAINT is the IPC maintenance utility that performs CA-ASM2 maintenance functions necessary for managing archived or backed up data sets. $MAINT removes logically deleted and expired archive or backup records from the IPC based on retention period processing or rules purging versions above an installation-specified limit. It also handles explicit tape processing requests.

$MIGRATE. The $MIGRATE utility migrates data sets from one set of disk volumes to another. It can be extremely helpful with the problem of quickly moving a massive number of data sets from one disk device to another. It may be the preferred way to move data in situations where many data sets need to be reblocked. It can reblock data sets to new block sizes compatible with the receiving volume. It also has the capability of downward reblocking of load module libraries without requiring the use of the linkage editor.

$OPTIONS. The $OPTIONS member of CAI.PARMLIB allows your information center to customize CA-ASM2 operations to the unique requirements of your environment. At the time CA-ASM2 is installed, your information center specifies various fields within $OPTIONS.

$PDM. This is the CA-ASM2 Physical Data Mover. It is a high-speed physical disk track read/write routine and operates without regard to normal data set attributes such as record format, block size and data set organization. It reads a track(s) up to a cylinder at a time. $PDM has fast track-image reads and writes because it ignores data set attributes; this is faster than reading one record at a time the way a logical data mover does.

$PDMUR. $PDMUR is an independent utility that can reload data sets that were unloaded with the CA-ASM2 Physical Data Mover, $PDM. For example, if a data set's IPC record was deleted but the data set still exists on tape, you can use $PDMUR to reload the data set. $PDMUR cannot reload data sets unloaded to the Disk Staging Area (DSA) or data sets unloaded to tape in compressed format.

$PDSUR. $PDSUR is a stand-alone version of the data mover program used to unload and reload partitioned data sets in CA-ASM2. $PDSUR is designed to be compatible with IEHMOVE. That is, $PDSUR uses the same unloaded format as IEHMOVE with the option of blocking up to 32K. Thus $PDSUR can reload data sets unloaded by IEHMOVE, and IEHMOVE can reload data sets unloaded with $PDSUR (blocked 800). $PDSUR cannot reload a data set that is spread across two or more tape volumes. It cannot restore a data set backed up with $PDM. It cannot reload data sets unloaded to the DSA or data sets unloaded to tape in compressed format.

$RSVP. $RSVP is a powerful interactive command that scans the VTOCs of your disk volumes, system catalogs, CA-1 Tape Management Catalog (TMC), CA-DYNAM/TLMS Volume Master File (VMF), CA-ASM2 Integrated Product Catalog (IPC), the SMS database, and input transaction files. $RSVP is a **command** whereas CA-RSVP is the name of the Report Selection and Variable Processing product under which the $RSVP command runs.

Index

Special Characters

$AR command
  BLIST 3-10
  SUBCOMMAND
    description 3-7
    example 3-21
$ASMBMON program 1-2
$BK command
  BLIST 3-10
  SUBCOMMAND 3-7
$OPTIONS 1-4
  IPC as input source 2-5
$RSBILL 5-4
$RSEXT1 exit 1-4, 2-35, 5-5
$RSEXT10 exit 5-5
$RSEXT11 exit 5-6
$RSEXT12 exit 5-6
$RSEXT1I 4-6, 5-4
$RSEXT2 exit 1-4, 2-35
$RSEXT2I 4-9, 4-15, 5-7
$RSEXT2O 4-9, 5-7
$RSEXT2R 4-8, 5-7
$RSOUT 5-2
$RSTRANS
  billing 4-5
    example 4-11
  DD statement 4-6
  JCL 5-2
  transaction files 3-3
$RSUD
  defining FILE default values 5-12
  defining other default values 5-16
  definition 5-10
$RSUEND
  defining 5-31
  definition 5-10
$RSUF
  defining fields 5-20
  definition 5-10
$RSUL
  defining lists of fields 5-25
  definition 5-10
$RSUSER
  reassembly 5-1
$RSUSER8 3-6, 3-24, 5-2, 5-9
$RSUV
  defining fields 5-27
  definition 5-10
$RSVOLTM 4-9, 5-3
$RSVP command
  batch 1-2
  description 1-2, 1-3
  fields A-26
  format 1-2
  preprocessing 3-2
  tailoring 2-38
  USED field 2-33
  using $RSEXT10 5-5
  VSAM data space 2-33
$SM command
  example 3-12, 3-14
  transaction file 3-12
&A 4-7
  description 4-7
&E 2-37
&I 2-37
&O 2-37
&R 4-7
  description 4-7
&U 4-7
  description 4-7
&V 4-7
  description 4-7
&X1 5-6
&X2 5-6
¬A
  example 4-11

¬R
example 4-11
¬U
example 4-11

A

ACTION keyword
description 3-4
example 3-12, 3-15, 3-17, 3-18, 3-19, 3-22, 3-26, 3-28
ADAM reports 2-24
ANDIFx
example 2-41
keyword description 2-15
operators 2-12
ANDx
example 2-41, 2-79, 2-80, 2-81, 2-82, 2-83, 2-86, 3-19
keyword description 2-14
operators 2-12
Archive
preprocessing 3-7
queue 3-7
Arithmetic symbols (syntax diagrams) 1-13
Ascending sort 2-17
ASM2CAT keyword
description 2-5
example 2-55, 2-74, 3-19
ASM2CAT selection examples 2-55
ASM2CMDU
transaction files 3-3
ASM2CMDU procedure 1-2
Asterisk
current run date 2-13
pattern masking 1-6
Attribute selection 2-12
AUXFLDS keyword 2-22

B

Backup
preprocessing 3-7
queue 3-7
Bad track locating 2-86
Batch mode 1-2
BEGIN keyword
description 2-10
example 2-42
Billing
$RSTRANS 4-6
Billing *(continued)*
command format 4-5
DD statements 4-6
description 4-2
fields A-21
history example 4-11
keyword description 4-5
one-step
account information 4-16
description 4-2
example 4-16
overview 4-1
rates example 4-11
two-step
description 4-2
example 4-11
flow chart 4-3, 4-4
BLIST keyword
billing 4-5
BLIST($SMCOMP example 3-15
BLIST($SMRLSE example 3-14
BLIST(ACTION1 example 3-12, 3-15, 3-17, 3-18, 3-19, 3-22, 3-24, 3-26
BLIST(ACTION2 example 3-24, 3-26
BLIST(BILLLIST example 4-11, 4-16
BLIST(SCRATCH example 3-28
example 2-78
transaction files 3-5
unregistered data sets 3-10
Boolean expressions/operators 2-12
BOTHEXITS keyword 2-35
BREAK keyword 2-29

C

CA-1
TMC fields A-24
CA-3 reports
ADAM101 2-24
ADAM102 2-24
ADAM102V 2-25
ADAM103 2-25
ADAM104 2-25
ADAM105 2-25
ADAM201 2-25
ADAM301 2-25
ADAM610 2-26
Available Free Space Report 2-25
Data Set Aging Status Report 2-25
Data Set Allocation Status Report 2-24
Data Sets Selected for Controlled Scratch Report 2-25

CA-3 reports *(continued)*
Detail Track Allocation on Volume Report 2-24
Migration/Backup Forecast Report 2-26
Nontemporary Data Set Status Report 2-25
Temporary Data Set Report 2-25
CA-ASM2
$DEFRAG component 2-87
command processor 2-33
obtaining information 2-1
CA-Dynam/TLMS
VMF fields A-25
CA-JARS 4-1
CA1TMC keyword
description 2-7
example 2-61, 2-62, 2-63, 2-64, 2-65
CATID keyword 2-6
Character values selection 2-9
CHARS keyword 2-22
CLUSTER keyword 2-32
Combined selection example 2-74, 2-76, 2-77
Comma
repeat symbol, use in 1-16
Command
continuation 1-3
format 4-5
$RSVP 1-2
archive queue 3-7
attribute selection 2-14
backup queue 3-7
data set name selection 2-10
output 2-21
report 2-21
sort 2-17
source selection 2-3
transaction files 3-3
unregistered data sets 3-10
user CSECT 5-9
user exits 2-35
VSAM elements 2-32
mode 1-2
processor RSVPCMDU 1-2
COMMENT keyword
example 3-12, 3-14, 3-15, 3-17, 3-18, 3-21, 3-22, 3-26, 3-28
queues 3-7
transaction files 3-4
CONTAINING keyword
description 2-10
example 2-42
Continuation character 1-3
control points 1-4
Controlled Scratch
ADAM301 report 2-25
CA-3 reports 2-25
CSECT 1-4
See also User CSECT
Customizing 1-4, 5-1
CYLS keyword 2-24

D

DASD
input selection 2-2
volume search 2-4
DASD Volume selection examples 2-39
DATA keyword
description 2-32
example 2-88, 2-89
Data set
attribute selection 2-12
extent fields A-20
fields A-20
pattern masks 1-6, 2-9
search 2-9
selection
ISPF panel 2-11
keywords 2-9
DATASPACE keyword 2-33
DATCLASS keyword
description 2-6
example 2-48
DDNAME keyword 2-22
Default values (syntax diagrams) 1-17
Defaults
field lengths A-19
keywords A-4
RSVP keywords 1-4
Defaults for keywords 1-4
Delimiters
syntax diagrams, use in 1-14
Descending sort 2-17
Disk Staging Area
listing data sets backed up to 2-59
listing data sets unloaded to 2-57
specific volume 2-60
DSNLEN keyword 2-22

E

ENDING keyword
description 2-11

ENDING keyword *(continued)*
example 2-42
EQ operator 2-12
Examples
archive queue 3-21
ASM2CAT selection 2-55, 2-57, 2-59, 2-60
backup queue 3-21
boolean logic 2-41
CA-1 selection 2-61
CA-ASM2 selection 2-39
CA-Dynam/TLMS selection 2-66
CA1TMC keyword 2-61, 2-62, 2-63, 2-64, 2-65
Combined selection 2-74, 2-76, 2-77
DASD volume selection 2-39, 2-41, 2-42, 2-44, 2-45
data set
attributes 2-41
name selection 2-42
descending sort 2-79
Disk Staging Area 2-57, 2-59, 2-60
IPC 2-55
listing users data sets' 2-39
listing VSAM activity 2-45
multiple input sources 2-74
one-Step billing 4-16
OSCAT selection 2-52, 2-53
output format 2-80
pattern masking 2-44
SMS
information 3-19
SMS selection 2-46, 2-47, 2-48, 2-49, 2-50, 2-51
sort 2-78
TLMSVMF keyword 2-66, 2-68, 2-70, 2-71, 2-72, 2-73
TMC selection 2-61, 2-62, 2-63, 2-64, 2-65
transaction files 3-12
two-step billing 4-11
unregistered data sets 3-22
VMF selection 2-66, 2-68, 2-70, 2-71, 2-72, 2-73
VSAM 2-45, 2-88
Executing $RSVP command 1-2
Exit router facility 5-5
EXIT1
keyword 2-35
sample input file 2-37
EXIT2 keyword 2-35
Exits
$RSBILL 5-4
$RSEXT1 1-4
$RSEXT10 5-5
$RSEXT11 5-6
$RSEXT12 5-6
Exits *(continued)*
$RSEXT1I 5-4
$RSEXT2 1-4, 5-7
control statements
&E 2-37
&I 2-37
&O 2-37
&X1 5-6
&X2 5-6
Exit 11 5-6
Exit 12 5-6
input file
for $RSEXT1 2-37
for $RSEXT1I 5-4
ISPF panel 2-38
post-sort 2-35
post-sort exit 5-7
router facility 5-5
RSVP 5-4
selection 2-35
Extents 2-85
EXTSORT keyword
description 2-19
example 2-78

F

Features summary 1-9
Fields
billing A-21
categories A-19
category A-19
data set A-20
data set extent A-20
default lengths A-19
defaults A-19
definitions A-26
descriptions A-26
IPC A-22
lengths A-19
literal A-21
miscellaneous A-21
SMS A-23
TMC A-24
VMF A-25
volume A-20
VSAM A-21
Free space (volume) 2-85, 2-86

G

GE operator 2-12
Glossary X-1
GT operator 2-12

H

Heading, default 2-20
HEADING1 keyword
  description 2-23
  example 2-51, 2-79, 4-11
HEADING2 keyword
  description 2-23
  example 2-79
Hyphen as command continuation 1-3

I

IEHPROGM
  delete data sets 3-24
  examples 3-22
  SCRATCH example 3-26, 3-28
  transaction files 3-3
  UNCATLG example 3-26
IF
  example 2-41, 2-55, 2-79, 2-80, 2-81, 2-82, 2-83, 2-85, 2-86, 2-88, 2-89, 3-18, 3-19, 3-28
  keyword description 2-14
  operators 2-12
INBILL keyword
  billing 4-5
  example 2-78, 3-22, 3-24, 4-11
  transaction data 2-6
  unregistered data sets 3-11
INBLIST keyword
  billing 4-5
  example 2-78
  INBLIST(BILLLIST example 3-22
  transaction data 2-6
  unregistered data sets 3-11
INDEXTOTALS keyword
  description 2-29
  example 2-84
Input selection/filtering phase 2-2
IPC
  fields A-22
  reporting example 2-55
  search 2-5
IPCCOML keyword 2-23
ISPF panels 1-2
  attribute selection 2-16
  Boolean Selection 2-16
  RSVP Data Set Selection 2-11
  RSVP Primary Input Source(s) Selection 2-8
  RSVP Report Format (output) 2-31
  RSVP Report Format (sort) 2-19
  RSVP User Exits 2-38
  RSVP VSAM Selection 2-34

K

KBYTES keyword 2-24
Keyword
  defaults 1-4
Keyword descriptions
  See also Keywords
  alphabetical list A-6
  archive queue 3-7
  attribute selection 2-14
  backup queue 3-7
  billing 4-5
  CA-1 2-7
  CA-Dynam/TLMS 2-7
  catalog 2-5
  DASD volumes 2-4
  data set name 2-10
  defaults 1-4, A-4
  exits 2-35
  IPC 2-5
  output format 2-22
  report format 2-22
  SMS database 2-6
  sort 2-18
  transaction
    data 2-6
    files 3-4
  unregistered data sets 3-10
  user CSECT 5-9
  user exits 2-35
  VSAM elements 2-32
Keywords
  ACTION
    description 3-4
    example 3-12, 3-15, 3-17, 3-18, 3-19, 3-22, 3-26, 3-28
  alphabetical list A-2
  ANDIFx
    description 2-15
    example 2-41
  ANDx
    description 2-14

Keywords *(continued)*
ANDx *(continued)*
example 2-41, 2-79, 2-80, 2-81, 2-82, 2-83, 2-86, 3-19
ASM2CAT
description 2-5
example 2-55, 2-74, 3-19
AUXFLDS 2-22
BEGIN
description 2-10
example 2-42
BLIST 4-5
BLIST($SMCOMP example 3-15
BLIST($SMRLSE example 3-14
BLIST(ACTION1 example 3-12, 3-15, 3-17, 3-18, 3-19, 3-22, 3-24, 3-26
BLIST(ACTION2 example 3-24, 3-26
BLIST(BILLLIST example 4-11, 4-16
BLIST(SCRATCH example 3-28
example 2-78
transaction files 3-5
unregistered data sets 3-10
BOTHEXITS 2-35
BREAK 2-29
CA1TMC
description 2-7
example 2-61, 2-62, 2-63, 2-64, 2-65
category list A-5
CATID 2-6
CHARS 2-22
CLUSTER 2-32
COMMENT
description 3-4, 3-7
example 3-12, 3-14, 3-15, 3-17, 3-18, 3-21, 3-22, 3-26, 3-28
queues 3-7
transaction files 3-4
CONTAINING
description 2-10
example 2-42
CYLS 2-24
DATA
description 2-32
example 2-88, 2-89
DATASPACE 2-33
DATCLASS
description 2-6
example 2-48
DDNAME 2-22
DSNLEN 2-22
ENDING
description 2-11

Keywords *(continued)*
ENDING *(continued)*
example 2-42
EXIT1 2-35
EXIT2 2-35
EXTSORT
description 2-19
example 2-78
HEADING1
description 2-23
example 2-51, 2-79, 4-11
HEADING2
description 2-23
example 2-79
IF
description 2-14
example 2-41, 2-55, 2-79, 2-80, 2-81, 2-82, 2-83, 2-85, 2-86, 2-88, 2-89, 3-18, 3-19, 3-28
INBILL 4-5
example 2-78, 3-22, 3-24, 4-11
transaction data 2-6
unregistered data sets 3-11
INBLIST 4-5
example 2-78
INBLIST(BILLLIST example 3-22
transaction data 2-6
unregistered data sets 3-11
INDEXTOTALS
description 2-29
example 2-84
IPCCOML 2-23
KBYTES 2-24
LASTLEVEL
description 2-11
example 3-18
LEVEL
description 2-5, 2-11
example 2-39, 2-42, 2-55, 2-74, 3-19
LIKE
description 2-11
example 2-44
LINES 2-23
LIST
description 2-24
LIST(CHARGDET example 4-11
LIST(CHARGES example 4-11, 4-16
LIST(MAP example 2-89
LIST(SPACE example 2-80, 2-81
LIST(VOLUME example 2-85, 2-86
listed
alphabetically A-2
by category A-5

Keywords *(continued)*
MBYTES 2-24
MERGE 2-18
MGTCLASS
description 2-6
example 2-49
MOUNT 2-4
NEWDATCLASS 3-4
NEWMGTCLASS 3-4
NEWSTRCLASS 3-4
NOEXITS 2-35
NOHEADING 2-23
NOMERGE 2-18
NOPRINT
description 2-29
example 2-78, 2-84, 3-22, 3-25, 4-11
NOSORT
description 2-18
example 3-24
NOTBEGIN 2-10
NOTCONTAIN 2-11
NOTENDING 2-11
NOTLAST
description 2-11
example 2-42
NOTLEVEL
description 2-11
example 2-42
NOTLIKE 2-11
NOTOTALS 2-30
NOTVOL 2-4
ORIFx
description 2-15
example 2-41
ORx
description 2-14
example 3-28
OSCATALOG
description 2-5
example 2-74
PERMANENT 3-8
PRINT
ADD 2-28
DEL 2-28
description 2-28
NEW 2-28
PRINT(ADD example 2-79, 2-81
PRINT(DEL example 2-41, 2-42, 2-44, 2-80, 2-85
PRINT(NEW example 2-45, 2-46, 2-47, 2-48, 2-49, 2-50, 2-51, 2-55, 2-74, 2-78, 2-83, 2-85, 3-19

Keywords *(continued)*
PRINT *(continued)*
PRINT(REP example 2-82
REP 2-28
QNAME 3-7
QUALIFY 3-7
RETPD
description 3-8
example 3-21
SMSNMLEN
description 2-29
example 2-46, 2-47, 2-48, 2-49, 2-50, 3-19
SORT
description 2-18
example 2-79, 3-22, 4-11
STGROUP
description 2-4
example 2-46, 2-47
STORGACD 2-6
STRCLASS
description 2-6
example 2-50
SUBCOMMAND
description 3-7
example 3-21
SUBTOTALS
description 2-30
example 4-11
TLMSVMF
description 2-7
example 2-66, 2-68, 2-70, 2-71, 2-72, 2-73
TOTALS 2-30
TRKS
description 2-24
example 2-42, 2-78, 2-79, 2-84, 2-85, 2-86
UGROUP
description 2-4
USERCSECT
description 5-9
example 3-24, 3-26, 3-28
VOLUME
description 2-4
example 2-39, 2-44, 2-45, 2-51, 2-74, 2-78, 2-80, 2-81, 2-82, 2-83, 2-84, 2-85, 2-86
VOLUME(* example 2-85
VSAMCAT
description 2-33
example 2-88, 2-89
Keywords (syntax diagrams) 1-13
Keywords and fields A-1

L

LASTLEVEL keyword
  description 2-11
  example 3-18
LE operator 2-12
LEVEL keyword
  description 2-5, 2-11
  example 2-39, 2-42, 2-55, 2-74, 3-19
LIKE keyword
  description 2-11
  example 2-44
LINES keyword 2-23
LIST keyword
  description 2-24
  LIST(CHARGDET example 4-11
  LIST(CHARGES example 4-11, 4-16
  LIST(MAP example 2-89
  LIST(SPACE example 2-80, 2-81
  LIST(VOLUME example 2-85, 2-86
Literal fields A-21
Literals
  L$ARDSN 3-4
  L$BKDSN 3-4
  L$RLSE 3-4
  L$RPVOL 3-4
  L$SMDSN 3-4
  LPAREN 3-4
  QUOTE 3-4
  RPAREN 3-4
Locating bad track 2-86
Logical expressions 2-12
LT operator 2-12

M

Macros 5-10
Masking 1-6, 2-9
MBYTES keyword 2-24
MERGE keyword 2-18
MGTCLASS keyword
  description 2-6
  example 2-49
Minus sign
  command continuation 1-3
  pattern masking 1-6
Miscellaneous fields A-21
MISSING - boolean expression 2-12
MOUNT keyword 2-4
Multiple
  input sources
    example 2-74
Multiple *(continued)*
  input sources *(continued)*
    reporting on 2-2
    searching 2-2
  line transactions 3-6
Multivalued fields 2-28
  field values A-1
  range of values 5-25
  sorting 2-18

N

NE operator 2-12
NEWDATCLASS keyword 3-4
NEWMGTCLASS keyword 3-4
NEWSTRCLASS keyword 3-4
NOEXITS keyword 2-35
NOHEADING keyword 2-23
NOLIMIT - boolean expression 2-12
NOMERGE keyword 2-18
NOPRINT keyword
  description 2-29
  example 2-78, 2-84, 3-22, 3-25, 4-11
NOSORT keyword
  description 2-18
  example 3-24
Not sign in pattern masking 1-8
NOTBEGIN keyword 2-10
NOTCONTAIN keyword 2-11
NOTENDING keyword 2-11
NOTLAST keyword
  description 2-11
  example 2-42
NOTLEVEL keyword
  description 2-11
  example 2-42
NOTLIKE keyword 2-11
NOTOTALS keyword 2-30
NOTVOL keyword 2-4

O

One-Step billing
  description 4-2
  example 4-16
Operators
  Boolean 2-12
  IF/ANDx/ORx/ANDIFx/ORIFx 2-12
ORIFx
  example 2-41
  keyword description 2-15

ORIFx *(continued)*
operators 2-12
ORx
example 3-28
keyword description 2-14
operators 2-12
OS catalog 2-5
OSCAT selection examples 2-52
OSCATALOG keyword
description 2-5
example 2-74
Output
default heading 2-20
format phase 2-20

P

Parentheses
syntax diagrams, use in 1-14
Pattern masking
asterisk 1-6
data set selection 2-9
description 1-6
minus sign 1-6
not sign 1-8
Performance 5-33
considerations 2-33, 5-33
SMS 5-33
VSAM 5-33
PERMANENT keyword 3-8
Phases
output format 2-20
record filtering 2-2
selection 2-2
sort 2-17
Points of control 1-4
Preprocessing
$AR example 3-21
$BK example 3-21
$SM 3-12, 3-14, 3-15
applications 3-1
archive and backup queues 3-7
definition 3-1
examples 3-22
field lists 3-5
IEHPROGM 3-24
overview 3-1
SCRATCH 3-24
SMS 3-1
SUBCOMMAND 3-21
transaction files 3-3
Preprocessing *(continued)*
TSO DELETE 3-17
UNCATLG 3-24
unregistered data sets 3-9
PRINT keyword
description 2-28
multivalued fields 2-28
PRINT(ADD example 2-79, 2-81
PRINT(DEL example 2-41, 2-42, 2-44, 2-80, 2-85
PRINT(NEW example 2-45, 2-46, 2-47, 2-48, 2-49, 2-50, 2-51, 2-55, 2-74, 2-78, 2-83, 2-85, 3-19
PRINT(REP example 2-82
Product changes 1-9
Programs
comma
repeat symbol, use in 1-16
parentheses
syntax diagrams, use in 1-14
punctuation
syntax diagrams, use in 1-13
Punctuation marks (syntax diagrams) 1-13

Q

QNAME keyword 3-7
QUALIFY keyword 3-7
Queue
archive 3-7
backup 3-7

R

Rates for billing example 4-11
Record filtering 2-2
Reporting 2-1
catalog 2-5
DASD volumes 2-4
examples 2-39
format phase 2-20
input selection/filtering phase 2-2
IPC
example 2-55
keyword descriptions 2-5
multiple input sources 2-2
report field list 2-24
SMS
database 2-4, 2-6
performance 5-33
sort order 2-17
source selection 2-2
space management 2-84, 2-85, 2-86

Reporting *(continued)*
transaction data 2-6
VSAM
elements 2-32
performance 5-33
Reports
Available Free Space Report 2-25
billing 2-26
Controlled Scratch 2-25
Data Set Aging Status Report 2-25
Data Set Allocation Status Report 2-24
Data Sets Selected for Controlled Scratch Report 2-25
Detail Track Allocation on Volume Report 2-24
Migration/Backup Forecast Report 2-26
Nontemporary Data Set Status Report 2-25
Temporary Data Set Report 2-25
Resource accounting 4-1
RETPD keyword
description 3-8
example 3-21
RSVPCMDU
executing $RSVP command 1-2
RSVPCMDU procedure 1-2

S

SCRATCH commands 3-6, 3-24
Scratch unregistered data sets 3-9
Selection
attributes 2-12
catalog 2-5
character values 2-9
DASD volumes 2-4
data set 2-9
data set attributes 2-12
data set name 2-9
input source 2-2
IPC 2-5
ISPF 2-8
pattern masks 2-9
SMS 2-6
transaction data 2-6
SMS
and RSVP 3-1
example
listing data classes 2-48, 2-51
listing data sets 2-47
listing management classes 2-49
listing storage classes 2-50, 2-51
listing storage groups 2-46
SMS *(continued)*
fields A-23
performance notes 5-33
selection 2-6
SMS selection examples 2-46
SMSNMLEN keyword
description 2-29
example 2-46, 2-47, 2-48, 2-49, 2-50, 3-19
Sort
ascending order 2-17
combine output 2-18
command format 2-17
descending order 2-17
keyword descriptions 2-17
multiple sources 2-18
multivalued fields 2-17
phase 2-17
SMS performance 5-33
VSAM performance 5-33
SORT keyword
description 2-18
example 2-79, 3-22, 4-11
Source selection
command format 2-3
ISPF 2-8
Space management
preprocessing 3-28
reporting 2-84, 2-85, 2-86
Specifying multivalued fields 2-28
Speed
access X-2
STGROUP keyword
description 2-4
example 2-47
STGROUPS keyword
example 2-46
STORGACD keyword 2-6
STRCLASS keyword
description 2-6
example 2-50
SUBCOMMAND keyword
description 3-7
example 3-21
SUBTOTALS keyword
description 2-30
example 4-11
Syntax diagrams
reading (how to) 1-13—1-18

T

Tailoring 1-4, 5-1
Tape Management Catalog
  See TMC
Terminology 1-5
TLMS fields A-25
TLMSVMF keyword
  description 2-7
  example 2-66, 2-68, 2-70, 2-71, 2-72, 2-73
TMC
  allocating 2-7
  DDNAME 2-7
  fields
    listing A-24
  searching 2-8
  searching fields 2-7
  selection examples 2-61, 2-62, 2-63, 2-64, 2-65
TOTALS keyword 2-30
Track locating 2-86
Transaction data search/selection 2-6
Transaction files
  $RSTRANS 3-3
  character field format 3-3
  command format 3-3
  generating commands 3-6
  keyword descriptions 3-4
  processing commands 3-3
TRKS keyword
  description 2-24
  example 2-42, 2-78, 2-79, 2-84, 2-85, 2-86
TSO
  command mode 1-2
  DELETE example 3-17, 3-22
  deleting data sets 3-24
Two-Step billing
  description 4-2
  example 4-11
TYPE field 5-6

U

UGROUP keyword
  description 2-4
Unregistered data sets 3-9
User CSECT 1-4
  assembling 5-31
  command format 5-9
  description 1-4, 5-1, 5-9
  keyword descriptions 5-9
  macros
    $RSUD 5-10
User CSECT *(continued)*
  macros *(continued)*
    $RSUEND 5-10
    $RSUF 5-10
    $RSUL 5-10
    $RSUV 5-10
  RSVP control 1-4
  tailoring 5-9
User exit control statements
  &A 4-7
  &R 4-7
  &U 4-7
  &V 4-7
  ¬A
    example 4-11
  ¬R
    example 4-11
  ¬U
    example 4-11
User exit descriptions
  $RSEXIT10 5-8
  $RSEXT1 5-4, 5-7
  $RSEXT10 5-5
  $RSEXT11 5-6, 5-8
  $RSEXT12 5-6, 5-8
  $RSEXT2 5-7
User list 3-11
USERCSECT
  $RSUSER8 3-6, 3-24, 5-2, 5-9
USERCSECT keyword
  description 5-9
  example 3-24, 3-26, 3-28

V

Variables (syntax diagrams) 1-13
VMF
  allocating 2-7
  fields
    listing A-25
  searching 2-8
  searching fields 2-7
  selection examples 2-66, 2-68, 2-70, 2-71, 2-72, 2-73
Volume
  fields A-20
  free space 2-85, 2-86
  search 2-4
  timestamp file $RSVOLTM 4-9
VOLUME keyword
  description 2-4
  example 2-39, 2-44, 2-45, 2-51, 2-74, 2-78, 2-79, 2-80, 2-81, 2-82, 2-83, 2-85, 2-86

VOLUME keyword *(continued)*
VOLUME(* example 2-84, 2-85
Volume Master Field
See VMF
VSAM
fields A-21
reporting 2-32
VSAMCAT keyword
description 2-33
example 2-88, 2-89